AFIT/GOR/ENS/00M-09

# Feature Selection for Predicting Pilot Mental Workload

THESIS

Julia A. East
2nd Lieutenant, USAF

AFIT/GOR/ENS/00M-09

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 074-0188 |
|---|---|---|---|



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | March 2000 | Master's Thesis |

**4. TITLE AND SUBTITLE**

FEATURE SELECTION FOR PREDICTING PILOT MENTAL WORKLOAD

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Julia A. East, Second Lieutenant, USAF

**7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)**

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GOR/ENS/00M-09

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Glenn Wilson, Ph.D., AFRL/HECP
2255 H Street, Building 33
Wright-Patterson AFB, OH 45433-7022
(937) 785-8748 Glenn.Wilson@wpafb.af.mil

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

Dr. Kenneth W. Bauer, Jr., Professor, AFIT/ENS
(937) 255-6565 x4328 Kenneth.Bauer@afit.af.mil

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**12b. DISTRIBUTION CODE**

**ABSTRACT (*Maximum 200 Words*)**

Advances in technology have the cockpits of the aircraft in the Air Force inventory increasingly complex. Consequently, mental demands on the pilot have risen. In some cases, mental demands were so overwhelming that pilots have forgotten basic flying techniques, such as G-straining maneuvers. The results have been fatal. Recent research in this area has involved collecting psychophysiological features, such as electroencephalography (EEG), heart, eye and respiration measures, in an attempt to identify pilot mental workload.

This thesis focuses on feature selection and reduction of the psycophysiological features and subsequent classification of pilot mental workload on multiple subjects over multiple days. A stepwise statistical technique and the signal-to-noise ratio (SNR) saliency metric were used to reduce the number of features required for classification. Factor analysis was used to compare the variables chosen by the discriminant procedure and the SNR metric as applied to a neural network. A total of 151 psychophysiological features were derived from data collected during an actual flight study. The original flight study contained three workload levels, low, medium and high. These levels were aggregated into two categories of pilot mental workload, low/medium and high. Mental workload associated with each flight segment was determined by difficulty of task.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| feature selection, neural networks, classification, pilot mental workload, feature reduction, discriminant analysis | 141 |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UNC |

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GOR/ENS/00M-09

# Feature Selection for Predicting Pilot Mental Workload

THESIS

Presented to the Faculty of the Graduate School of Engineering and Management
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Julia A. East, B.S.
2nd Lieutenant, USAF

March 21, 2000

AFIT/GOR/ENS/00M-09

# Feature Selection for Predicting Pilot Mental Workload

Julia A. East, B.S.

2nd Lieutenant, USAF

Approved:

| | |
|---|---|
| Dr. Kenneth W. Bauer Jr.<br>Advisor | 7 MAR 00<br>Date |
| Maj Jeffrey W. Lanning<br>Reader | 7 Mar 00<br>Date |

*Acknowledgements*

I would not have completed this thesis without the support of many people. First I would like to thank my advisor Dr. Kenneth Bauer. Not only is he one of the most knowledgeable people I know, he was also a calming influence throughout my research effort. Not only did he lead me through this research effort, by he also led me through lessons on writing, which improved drastically throughout the 6 months we worked together. I would also like to thank my reader, Maj Jeffrey Lanning. His corrections were numerous, but they also helped me become a better writer. Many thanks go out to all my friends that had to suffer through this research effort with me. A special thanks goes Ken, who stuck by me through thick and thin throughout our program here at AFIT. Thanks also to Mike, Bruce, Rene and Dave who have always been there for support and help when I needed it. I would be remiss if I did not thank Lt Col Stephen Alsing. He never tired of me asking questions on Matlab, neural networks, or Scientific Workplace. I will never be able to thank him for all the Matlab code he provided me with. This research effort never could have left the ground without the help of AFRL/HE. I would like to thank all the guys at the lab: Dr. Wilson, who helped me understand exactly what psychophysiological meant and the workings of the human brain; Chris Russell, who provided endless Matlab help; and Jared Lambert, the main data man. Last but not least I would like to thank my family and friends for their support and understanding. Thanks to my Mom, Dad, Kathy, Charity, Pete and Sandra.

Julia A. East

*Table of Contents*

## *List of Figures*

## *List of Tables*

*Abstract*

. As advances in technology are made, the cockpits of the aircraft in the Air Force inventory have become increasingly complex. Consequently, mental demands on the pilot have risen. In a worst case scenario, the pilots have been so saturated with inputs they have actually forgotten to carry out the fundamentals of flying, such as G-straining maneuvers, resulting in several fatalities. Recent research in this area has involved collecting psychophysioloical features, such as electroencephalography (EEG), heart, eye and respiration measures, in an attempt to identify pilot mental workload. This thesis focuses on feature selection and reduction of the psychophysiological features and subsequent classification of pilot mental workload on multiple subjects over multiple days. A stepwise statistical technique and the signal-to-noise (SNR) saliency metric were used to reduce the number of features required for classification. Factor analysis was used to compare the variables chosen by the discriminant procedure and the SNR saliency metric as applied to a neural network. A total of 151 psychphysiological features were derived from data collected in an actual flight study. The original flight study contained three workload levels, low, medium and high. These levels were aggregated into two categories of pilot mental workload, low/medium and high. Mental workload associated with each flight segment was determined by difficulty of the task in conjunction with subjective measures from the pilots that participated in the study.

# Feature Selection for Predicting Pilot Mental Workload

## *I. Introduction*

### *1.1 Statement of Problem*

This research continues the effort to use artificial neural networks and statistical classifiers to classify pilot mental workload. This thesis expands upon previous work [11, 14] using multivariate discriminant models and feedforward multilayer neural networks to classify mental workload using data collected from an actual flight. One proposed research question is: Can we construct a classifier that is robust enough to account for individual variations from day to day? Stated in other words, is one net sufficient to predict day to day? A second and perhaps more interesting question is: Can we form one classifier that is robust enough to account for variations between pilots? Studies were conducted using data obtained on two individual pilots, each pilot flying on two different days.

Many elements go into determining the answer to the stated research questions. The first element that must be considered is the development of a parsimonious set of salient input features into a classifier [14]. Screening techniques are used to reduce the number of input features while still maintaining the power to accurately classify. More questions are raised when considering different screening techniques. Does this set of input features differ depending upon which screening technique is used? Do the input features remain the same from day to day? Do the input features remain the same when predicting across pilots? The input features considered in this research are psychophysiological features to include brain electric activity,

heart rate, respiration, and eye blink measures. Another element to consider is the classification accuracy. Does one classifier consistently produce better results? If so, how robust is this classifier to "messy" input data?

### *1.2 Background*

The advanced fighter jets the Air Force currently uses today are technologically mindboggling compared with the early reconnaissance craft used in the first World War. Along with this technological advancement comes a greater demand on the operator of the craft. As technology advances it is incorporated into the cockpits of Air Force aircraft. With this incorporation comes the need for the pilot to split his attention between many different tasks. When this attention gets divided and the pilot gets into a stressful or mentally demanding state, a potential for mental overload presents itself. One of the most devastating examples of mental overload is found in studies on pilots of fighter aircraft. Pilots have become so involved in trying to pay attention to everything that is happening they forget to perform basic tasks, such as G-force straining maneuvers. As a result, pilots have lost consciousness and consequently lost their lives. One pilot was so concerned about this matter, he conducted a personal study after losing consciousness due to G-forces himself [2]. His study revealed that over a period of 10 years there were 14 GLOC(G induced loss of consciousness) incidents. All but one of these occurred during mentally demanding portions of flight. This mental overload was the only common factor in all 13 cases. In order to save pilot lives, an effort is being made to create an advanced warning system to notify the pilot of a potential mental overload.

The Air Force Research Laboratory(AFRL)/Human Effectiveness Directorate (HE) at Wright-Patterson Air Force Base, Ohio is one of the leading research facilities in mental workload analysis [1]. AFRL/HE has conducted numerous studies using physiological features to determine mental workload. The physiological features determined most influential in classifying workload level are: brain electrical

activity, heart rate, breath rate, and eye blink measures [13, 28–32]. These features have been determined in various laboratory, simulator and flight settings. Current AFRL/HE efforts involve the Wright-Patterson Aero Club flying Piper Cubs. Data was collected using 10 pilots flying a specified route on two separate days. The pilots wore special equipment to monitor and record brain electrical data, heart rate, breath rate and eye blink measures. No known research has ever been done to see if a classifier constructed using data from one day will yield acceptable results trying to predict mental workload using data from a second day. Similarly, no known research has been done to examen how a classifier will perform trying to predict across pilots.

Previous research has used feedforward multilayer perceptron (MLP) neural networks to classify workload level using flight simulator data. Studies have shown that physiological features vary in importance in laboratory settings versus flight settings [29]. If accurate flight classifications are to be made, actual flight data must be used. The differences in the laboratory and flight data suggest that several physiological measures must be included to accurately classify mental workload in multitask, mentally demanding situations such as flight.

Neural networks are inspired by the workings of the human brain. Inputs into the net are weighted according to importance, causing the net to classify the input data into any number of output states. This type of classification could be used in the cockpit to classify pilot mental workload. If the classification of input data can be practically implemented into the cockpit for everyday use, the potential to forewarn an operator of potential overload could save lives.

### *1.3 Research Objectives*

Flight data has been gathered concerning pilot workload. As mentioned before, classification efforts thus far have concentrated on using simulator data. The next step is to classify the flight data using the same analysis procedures as were used for the simulator data (statistical and MLP classifiers). The flight data contains a lot of

input information. Investigations are done using screening methods to reduce this set of input features to a manageable set that yields adequate accuracy, yet not be overwhelming. While using the entire set of input features may give a more precise picture of the mental state the pilot is experiencing, it could be very time consuming to use all of the features to classify workload levels. It is also important to keep in mind that some warning system is going to be placed in an aircraft. The system has to be small enough and fast enough to notify the pilot of mental overload while the pilot still has time to react. By cutting down on the number of input features required, the warning system will be that much faster as well as more practical. So far, discussion has focused on screening out the input features for one set of data, one pilot. It will also be helpful to obtain a set of input features that holds for all pilots, not just one, so a standardized system can be integrated into the cockpit.

After a set of features is chosen as input for the neural network, the data is broken into different sets for training of the network. Typically a training set, a test set and a validation set are created. The first set is used to train the network, the second to test that the network has been adequately trained and the third to validate that the network actually works the way it is supposed to. The second set of data, the test set, is crucial to creating a neural network. This data set allows the creator of the network to avoid the possible problem of overfitting the neural network.

### *1.4 Research Methodology*

The first step in this research takes the raw data and preprocess it into a usable form. This is done using Fast Fourier Transforms (FFTs). Much of the data is continuous or near continuous. It has to be cut into more manageable segments to be input into a classifier. The second step takes the processed data and develops a parsimonious set of input features. There are several saliency measures available for this task. The saliency measures considered in this thesis are a discriminant stepwise

screening method run in the statistical program *SAS,* and the signal to noise ratio (SNR) saliency measure. The SNR saliency metric is the chosen saliency measure in Laine's thesis [4, 14]. Once that we have our set of input features, the data is analyzed two different ways. One way is using multivariate discriminant analysis. This gives us a statistical classifier method. The multivariate analysis includes both linear and quadratic classifiers. The second way involves a feedforward MLP neural network. A comparison of classification accuracies will be done between both classification methods.

### *1.5 Results*

The results of this research could bring the Air Force one step closer to saving one of our greatest resources – our pilot's lives. At the very least it will give an idea as to which classification model will be most practical to implement in the assessment of pilot mental workload.

# *II. Background and Literature Review*

## *2.1 Introduction*

The background and literature review gives detailed information on all subjects pertinent to this research effort. The first section gives a brief background on neural networks. The second section concentrates on feedforward multilayer perceptron neural networks, while the third section focuses on saliency screening methods for the input features. The fourth section touches on multivariate analysis. Finally, the last section gives a detailed description of the psychophysiological features considered as inputs for this research.

## *2.2 Neural Networks - Background*

Neural networks are inspired by the workings of the brain. The brain is composed of a network of neurons [18]. A neuron in the brain receives input from other neurons. This causes the neuron to fire and send signals to other neurons in the chain. This is the basis of how we learn. As our experiences grow, connections between neurons strengthen and weaken. The neurons are not "aware" of what has happened as a whole, they are only capable of responding in a certain manner when that situation or a similar situation presents itself. This pattern of learning is the basic principle that the neural network is built upon. In the 1940's, Warren McCulloch and Walter Pitts first explored the computational capabilities of networks by creating a network made of model neurons [17]. Their simple model had the neuron fire when the sum of its inputs exceeded a threshold. They thought that models of this type not only appropriately modeled symbolic logic, but were also adequate for modeling perception and behavior.

In the 1950's, a man named Frank Rosenblatt voiced his concern about models like McCullogh and Pitts' [18]. He thought these types of models to be unbiological. They required precise connections and timings and didn't take into account the

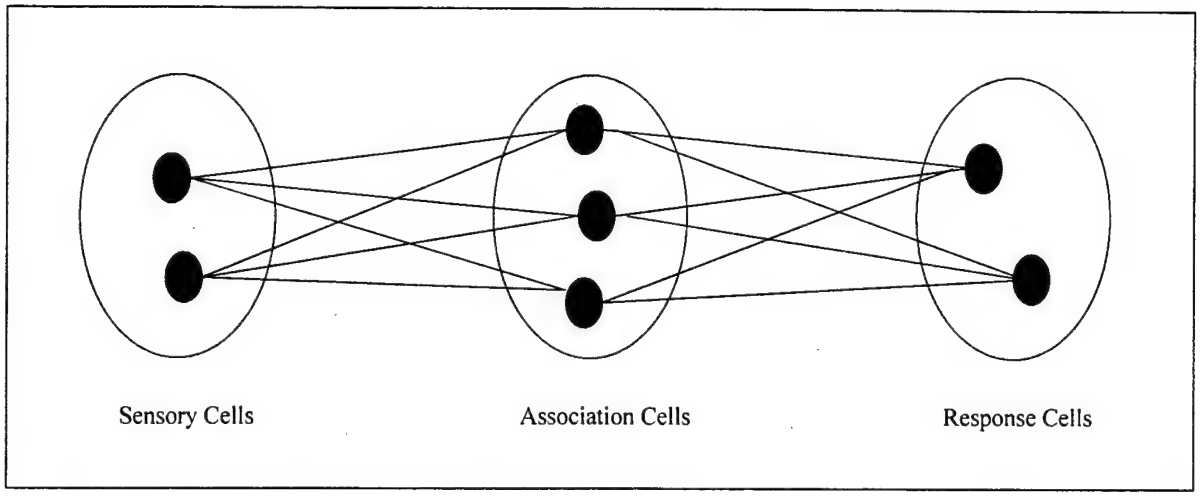


Figure 2.1 Perceptron

unpredictability and randomness of a real biological system, namely a real neural network [18]. Rosenblatt's idea was to create a neural network able to distinguish between similar and different experiences. This approach resulted in the creation of Rosenblatt's perceptron. The simplest version of this perceptron is formed of three layers, shown in Figure 2.1.

The first layer consists of sensory cells. The sensory cells are connected, on a random basis, to the next layer that contains the association cells. These association cells are in turn connected, again in a random fashion, to response cells in the third, or response layer. These response cells produce the output of the network. This new idea of a "perceptron" began the process of accurately assessing the true nature of mental functions. Although it was on the right track to accurately portraying true mental functions, the perceptron was limited to learning how to classify linearly separable functions. If the region was not linearly separable, such as the exclusive-or (XOR) problem, as shown in Figure 2.2, the perceptron could not correctly classify all cases. A way had to be found to correct this problem.

Researchers continued to work on different network designs in an attempt to solve this problem. Finally in 1986, over 30 years after the perceptron was first con-

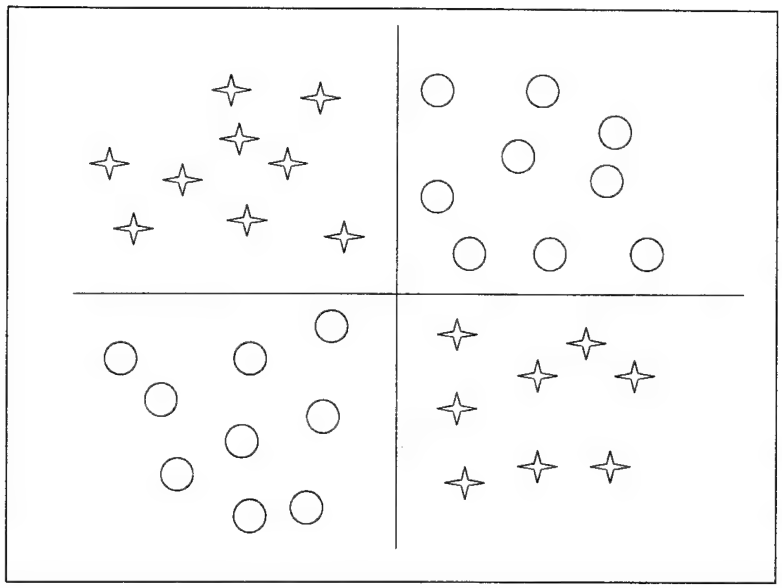

Figure 2.2 XOR Classification Problem.

ceived, three researchers, David Rumelhart, Geoffrey Hinton, and Ronald Williams announced the discovery of a method for allowing networks to discriminate between classes for nonlinearly separable regions. Their "backward propagation of errors" method, or backpropagation, led to modern day neural networks. Backpropagation is simply a gradient search method on the error surface produced after training. The goal is to minimize error. That is, to get the network to classify accurately as often as possible. The ability to adaptively minimize error makes the neural network a highly used tool in classification efforts today

The basic network used in this research is the feedforward multilayer perceptron neural network, shown in Figure 2.3. There are three layers to the network: the input layer; the hidden node layer; the output layer. Inputs, typically of various orders of magnitude, are fed into the network via the input nodes. These nodes pass inputs via a weighted branch to hidden layer nodes. The hidden nodes then calculate the weighted sum of all inputs received and sends this sum through an activation function. In order for the network to consider all inputs equally, the activation function squashes the inputs into a small range. Modified inputs are now

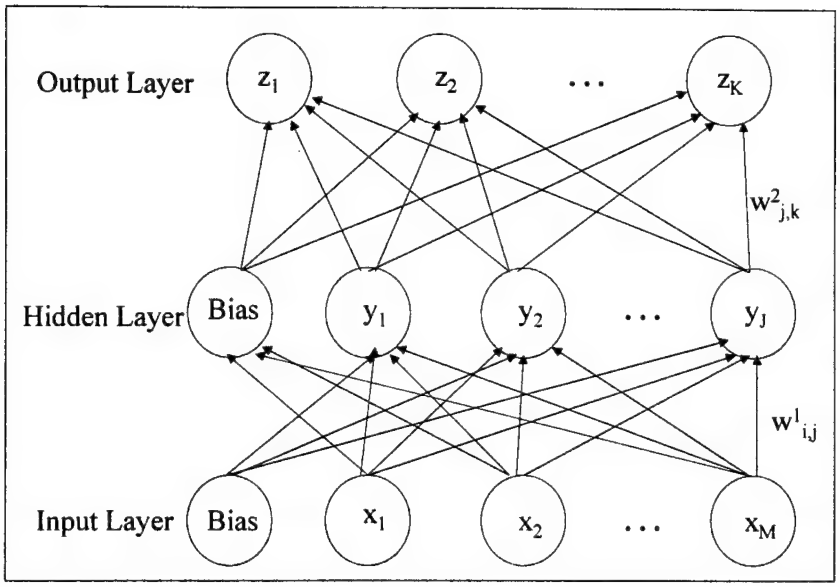


Figure 2.3 Multilayer Perceptron Neural Network with Bias

sent from the hidden node layer to the output layer via weighted branches and an output is produced. As seen in Figure 2.3, the flow is entirely in a forward direction. There are no cycles back to the input nodes hence the name feedforward multilayer perceptron (MLP) neural network.

*2.2.1 Definitions.* Before proceeding, it will be helpful to establish some basic definitions.

- **Activation function** - a mathematical function that takes the weighted activation values entering a unit, sums them, and translates the result to a position along a given scale. The activation function will often squash the summed value to a specified range (typically, 0 to 1, or -1 to 1) and is consequently also known as the squashing function [22].
- **Weights** - connections of varying strength that carry activation information between network units [22]

- **Backpropagation** - method that uses a gradient descent along the error surface to find the optimal values for the weights [8, 11, 18, 22]
- **Epoch** - one presentation of the entire training set to a neural network [22]
- **Batch mode learning** - the entire training set is presented, a single average error value is calculated, and the network is updated once according to that average error [22]
- **Feedforward neural network** - a neural network in which the flow of activation is in a single direction [14]
- **Momentum** - method that improves the training time of the backpropagation algorithm while enhancing the stability of the process (helps in avoiding local minima in the error surface) [22, 26]
- **Learning rate** - used with momentum to enhance the backpropagation algorithm by telling the network how slowly to progress (avoids jumping over the solution with momentum) [22]
- **Sigmoid activation function** - an activation function that squashes its input into a range, typically from 0 to 1 or from 1 to -1 [22]

## *2.3 Feedforward Multilayer Perceptron Neural Network*

The basic construction of the MLP neural network was presented in the last section. The next step is to delve deeper into the inner workings of the neural network and examine it from input nodes, weight connections, hidden layer nodes, and finally the output nodes.

*2.3.1 Input Nodes.* Each training set presented to a neural network will enter the network via input nodes. Some amount of preprocessing is generally required on the input data. One common preprocessing step is the standardization of the data before presentation to the neural network. The standardization will take

out any bias that may be caused by individual units of the inputs. When presented with data each input node takes one feature that helps determine the output of the network. For example, suppose a network has been constructed to determine whether or not it is likely to rain. Let's say that the inputs for this network are temperature, weather forecast (barometric pressure), and the amount of rain that historically falls that time of the year. The network will perform calculations on this set of inputs to determine whether or not it is likely to rain that day. The inputs in this example all have different units. The temperature input is going to be quite a bit larger than the amount of rain input. Preprocessing will standardize the data. This enables the neural network to consider each input equally. In addition to the input nodes for each feature, the input layer also contains a bias that is typically set to 1.0. The purpose of this bias is to set inputs into the correct range for the next layer's activation function. This is true whether the next layer is the hidden node or the output layer. Data leaves the input layer via a series of connecting weights that bring the data to the hidden node layer.

*2.3.2 Weight Initialization.* If the network has never been used before (no backpropagation has been performed) it is necessary to assign initial values to the weight connections leading from the input nodes to the hidden layer nodes and from the hidden nodes to the output layer. There is a smart way to assign initial weights that will give the best possible start for the network to begin learning. The quickest way to begin is to set initial values such that the weighted sum of inputs into the next layer is close to zero for every node, regardless of input [18]. This weighted sum is fed into an activation function in the next layer. The activation function(often the logistic activation function, discussed later) causes the output of that node to be close to 0.5. The desire for the output to be close to 0.5 is two-fold. First, we don't know what the actual output of the node should be. A midrange value is the safest to start with. In the case of the output node, a midrange value minimizes the squared error. This is good since our goal with the neural network

is to minimize the error between classifications of the neural network and the actual classifications. Second, we want to avoid extreme output values because they will have very small error derivatives [18]. Error derivatives are the foundation of the backpropagation algorithm. If the error derivatives are small, the weight changes are small, thus learning could take a long time. A midrange output of 0.5 is as far from the extremes as we can get.

Notice in Figure 2.3 that there are weighted connections from each input node to each hidden node. The theory behind this is that every input will have some effect on the hidden layer nodes. The weights determine the size of this impact. The accepted strategy is to make the initial weights very small for the first layer. A starting range of -0.05 to 0.05 is typically used [11, 14]. Backpropagation adjusts these weights to the correct values as determined by the neural network.

Weighted connections also lead from all hidden layer nodes to the output nodes. As stated earlier, having weights connecting input and hidden layer nodes close to zero causes the output of the hidden nodes to be near zero. Half the weights connecting the hidden layer nodes and output nodes should have weights set to 1 and the other half to -1. If there are an odd number of nodes, the bias weight should be set equal to 0. This ensures the output nodes generate values close to the midrange of the activation function.

There is one important factor to keep in mind when initializing the weights. Never set all the weights equal to each other. If all the weights in the first layer are equal, all hidden nodes see the exact same input, and produce the exact same output. Thus the contribution to error is the same across all the sub-networks. Since a network learns based on the error derivatives, if they are all the same, all weight adjustments will be the same and we fall into a vicious cycle. As a result, the network will be unable to solve a nonlinear problem.

*2.3.3 Hidden Nodes.* There are not a set number of hidden nodes that go in the hidden node layer. If too few hidden nodes are included, the network may not be able to solve the problem. That is, the network will be unable to correctly classify data sets presented to it. If the network contains too many hidden nodes, overfitting of the data can occur. The concept of overfitting will be discussed in depth later, but it basically means that the network will only be able to solve for the data it is trained on and lose flexibility in accounting for data that may be slightly different. While there is no known algorithm for deciding the number of nodes that should be added to the hidden layer of the neural network, it has been shown that a single hidden layer is sufficient to approximate any response surface, as long as it contains an adequate number of hidden nodes [11, 14]. There are a few algorithms designed to set upper limits on the number of hidden nodes required for a neural network. Kolmogorov's theorem is one such algorithm. Kolmogorov's theorem proves the upperbound for the number of hidden nodes will never be more than twice the number of input nodes [22]. A separate theory on the number of hidden nodes is presented in Steppe's work [20]. Her upperbound on the number of hidden nodes is shown in the following equation.

$$HN < \frac{0.5P - 1}{M + 1} \tag{2.1}$$

where $P$ is the number of exemplars and $M$ is the number of features presented as inputs into the model. This equation works well unless the number of features is large and the number of exemplars is small. In these situations Equation 2.1 may underestimate the number of hidden nodes necessary to handle the complexity of the problem. Both Kolmogorov's and Steepe's methods for hidden node selection are only heuristic techniques. The selection of the number of hidden nodes is very much an art form and depends on the complexity of the problem at hand.

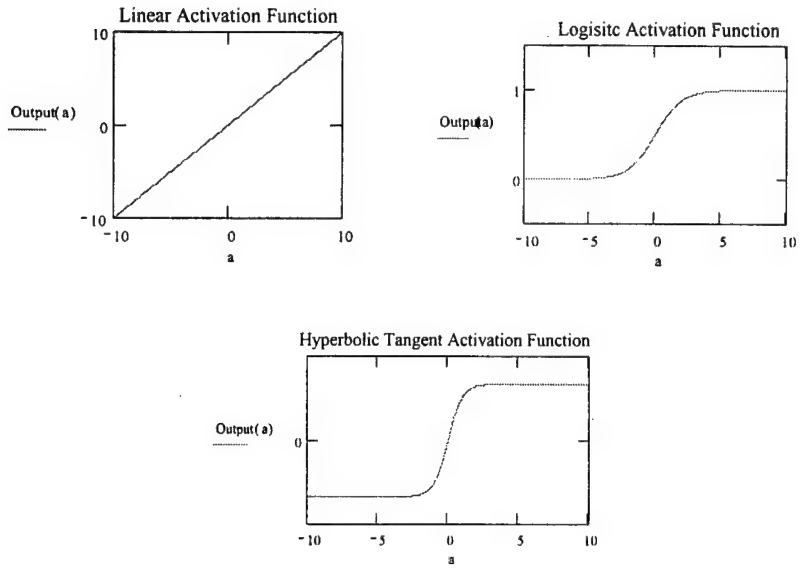


Figure 2.4 Activation Functions.

*2.3.4 Activation Function.* Each hidden node receives a weighted sum of the inputs. At each node, this weighted sum passes through an activation function. Let's consider three different activation functions: the linear, the logistic and the hyperbolic tangent (Figure 2.4).

Linear activation functions are only used when the data set is known to be linearly separable. The linear activation function is simply:

$$f(a) = a \tag{2.2}$$

with a derivative of:

$$f'(a) = 1 \tag{2.3}$$

While the linear activation function benefits from its simplicity, it is unable to handle data sets that are not linearly separable. The ideal activation function is a sigmoid (S-shaped) [11,18,22,26]. The sigmoid is chosen to account for the "noise saturation

dilemma" [26]. When inputs have varying magnitudes, we encounter the noise saturation dilemma. For example, going back to the sample inputs for our rain example, temperature could be 45 $^0$F, while typical amount of rain at that time of year could be something small, like 0.01 inches. A sigmoid function will take care of the input magnitude difference by putting small values in the central area of the sigmoid function (around zero) and larger near the extremes of the function. As the input into the activation function grows in magnitude, it is squashed into smaller parts of the extremes.

There are three desirable characteristics of the sigmoid function. It is bounded (by 0 and 1 or -1 and 1), monotonically increasing, and differentiable everywhere. This last characteristic is important because the backpropagation algorithm requires that the activation function be differentiable everywhere. There are many sigmoid functions, however, the most commonly used is the logistic function:

$$f(a) = \frac{1}{1 + e^{-a}} \tag{2.4}$$

The logistic function squashes the input into the range from 0 to 1. The first derivative of logistic function is very simple to calculate. It is given as:

$$f'(a) = f(a)[1 - f(a)] \tag{2.5}$$

Since, the input into every hidden node is a weighted sum of the inputs, we know the value of a, the input into the logistic function.

$$a_j = \sum w_{ij} I_i + Bias \tag{2.6}$$

where

$a_j$ = summed weighted inputs(including bias term) entering the activation function

$w_{ij}$ = weight from node $i$ to node $j$

$I_j$ = input from node $i$

Equation 2.6 holds true at the hidden nodes, as well as the output nodes. While the logistic function is seen most frequently, other sigmoid functions are also used. One example is the hyperbolic tangent:

$$f(a) = \tanh(a) = \frac{e^{-a} - e^{a}}{e^{a} + e^{-a}} \tag{2.7}$$

The hyperbolic tangent squashes the input into the range of -1 to 1 as opposed to the 0 to 1 range given by the logistic function. The first derivative is easy to calculate for tanh(a):

$$f'(a) = 1 - [f(a)]^2 \tag{2.8}$$

So far we have journeyed from input nodes, through the first set of weights, to the hidden nodes. At the hidden nodes, we saw how the sum of the weighted inputs is processed through an activation function. These values are next sent to the output nodes via weighed branches. At the output nodes, similar calculations are made with another activation function.

Now we have a complete picture of the MLP neural network that we will be working with as well as its initial settings. It is now time to concentrate on how a MLP neural network actually learns. The algorithm that drives the learning process for a neural network is called backpropagation.

*2.3.5 Backpropagation.* In general, backpropagation is a way to find the optimal values of the weights that connect our neural network together. Inputs are fed into the neural network via the input nodes. They travel along the weight branches to the hidden layer nodes and then through more weight branches to the output nodes. The idea is to train the network before using it for actual classification. A training set with known class membership is presented to the network. After the initialization phase, the training set will be sent through the network. The output generated by the network is then compares with the desired output. The difference between these two values defines an error surface for the problem. This error determines how the weights are going to change. Backpropagation uses a gradient descent to find the minimum error on the error surface.

Initially, all data received should be separated into three different sets: training, training-test, and validation. Allocation of the data to each set will be discussed in detail in a later section. The training set is used to train the neural network. An input vector $\mathbf{x}^p$ is randomly selected for input to the neural network. The input vector $\mathbf{x}^p$ is the $p$th vector of the training set. This input vector is now sent through the network where the weights are set at the settings mentioned in Section 2.3.2. The instantaneous output error, $\varepsilon_o^p$, associated with $\mathbf{x}^p$ is calculated using the $p$th vector of observed outputs, $\mathbf{z}_k^p$, and the corresponding vector of desired outputs, $\mathbf{d}_k^p$. In this context $p$ represents the $p$th input vector of data, and $k$ represents the number of output nodes. The number of output nodes typically equals the number of classes. The calculation for instantaneous error, $\varepsilon_o^p$, is given as:

$$\varepsilon_o^p = \sum_{k=1}^{K} (d_k^p - z_k^p)^2 \tag{2.9}$$

where

$d_k^p$ = the desired output vector associated with the $p$th input vector

$z_k^p$ = the observed output vector associated with the $p$th input vector

$K$ = the number of output nodes

Now that the error surface is defined, the next step is to follow gradient along the path of steepest decent. This path is determined by taking the partial derivative of the error surface, $\varepsilon_o^p$, with respect to our weights. The calculations for the partial derivative of the error surface, $\delta$, depends on which layer of weights are being considered. The following calculations show $\delta$ for both the hidden layer to output layer weights and the input layer to hidden layer weights. For the hidden layer to output layer weights case we use,

$$\delta_k^2 = (d_k^p - z_k^p) z_k^p (1 - z_k^p) \tag{2.10}$$

while in the input layer to hidden layer weights case we use,

$$\delta_k^1 = x_j^1 (1 - x_j^1) \sum \delta_k^2 (w_{jk}^2)^{old} \quad \text{for } k = 1, ..., K \tag{2.11}$$

where

$(w_{jk}^2)^{old}$ = the old weight from hidden node j to output node k

It is important to note that the partial derivatives for the error surface take on these equations if the activation functions in the network are sigmoid functions. If the activation functions are linear, the partial derivatives take on a slightly different form.

Hidden layer to output layer weights(linear activation function):

$$\delta_k^2 = (d_k^p - z_k^p) \tag{2.12}$$

Input layer to hidden layer weights (linear activation function):

$$\delta_k^1 = \sum \delta_k^2 (w_{jk}^2)^{old} \quad \text{for } k = 1, ..., K \tag{2.13}$$

After finding the gradient decent direction, these partial derivatives can be used to update the weight parameters in the network. The weight updates are given as: Hidden layer to output layer weights:

$$(w_{jk}^2)^{new} = (w_{jk}^2)^{old} + \eta \delta_k^2 x_j^1 \tag{2.14}$$

Input layer to hidden layer weights:

$$(w_{ij}^1)^{new} = (w_{ij}^1)^{old} + \eta \delta_j^1 x_i^p \tag{2.15}$$

where

$(w_{jk}^2)^{new}$ = the updated weight from hidden node j to output node k

$(w_{jk}^2)^{old}$ = the old weight from hidden node j to output node k

$(w_{ij}^1)^{new}$ = the updated weight from input node i to hidden node j

$(w_{ij}^1)^{old}$ = the old weight from input node i to hidden node j

$\eta$ = the learning rate, or the training step size

$x_i^p$ = the $i$th input feature of the $p$th input vector

$x_j^1 = f(\sum w_{ij}^1 x_i^p)$ - the output of hidden node j $(i = 1, ..., M)$

In the equations above, a new parameter, $\eta$, was introduced. This is the learning rate, or the training step size for the neural network. It typically takes on values between zero and one. The learning rate dictates the proportion of error that will be used to update weights during backpropagation. It is a balance between learning speed and stability of the system [22]. The size of $\eta$ controls how fast the network learns. If $\eta$ is large, the network learns faster, however there is a chance of getting stuck oscillating around a local minimum in the error surface. In this case the network won't improve much as more training vectors are presented to it. If $\eta$ is small there is less chance of oscillating around a local minimum, however there is increased computational time. Additionally, if $\eta$ is too small, there is the chance of getting stuck in a local minimum and missing the true minimum of the error surface. A typical learning rate value is $\eta$= 0.25 [11, 14, 22].

*2.3.5.1 Momentum.* Momentum in the backpropagation algorithm can be helpful in speeding up convergence and avoiding any local minima in the error surface [27]. The concept behind momentum is to make more conservative changes in the weights. Momentum causes the weight changes to be affected by the size of the previous weight changes [22]. As a consequence, a new term, $\alpha$, is introduced as the momentum term in our weight updating equations. This momentum parameter is a constant that determines the effect of past weight changes on the current weight change [14]. The new weight update equations, with the momentum term, are given as: Hidden layer to output layer weights:

$$[w(t+1)^2_{jk}]^{new} = [w(t)^2_{jk}]^{old} + \eta\delta^2_k x^1_j + \alpha\Delta[w(t-1)^2_{jk}]^{old,old} \tag{2.16}$$

Input layer to hidden layer weights:

$$[w(t+1)^1_{ij}]^{new} = [w(t)^1_{ij}]^{old} + \eta\delta^1_j x^p_i + \alpha\Delta[w(t-1)^1_{ij}]^{old,old} \tag{2.17}$$

where

$$
\begin{aligned}
[w(t+1)^2_{jk}]^{new} &= \text{the updated weight at epoch t+1 from hidden node j to output node k} \\
[w(t+1)^1_{ij}]^{new} &= \text{the updated weight at epoch t+1 from input node i to hidden node j} \\
[w(t)^2_{jk}]^{old} &= \text{the old weight at epoch t from hidden node j to output node k} \\
[w(t)^1_{ij}]^{old} &= \text{the old weight at epoch t from input node i to hidden node j} \\
t &= \text{the training epoch} \\
\alpha &= \text{the momentum term}
\end{aligned}
$$

and

$$
\begin{aligned}
\Delta[w(t-1)^2_{jk}]^{old,old} &= [(w(t)^2_{jk})^{old} - (w(t-1)^2_{jk})^{old,old}] \\
&\quad \text{weight change from epoch t-1 to epoch t} \\
\Delta[w(t-1)^1_{ij}]^{old,old} &= [(w(t)^1_{ij})^{old} - (w(t-1)^1_{ij})^{old,old}] \\
&\quad \text{weight change from epoch t-1 to epoch t}
\end{aligned}
$$

Notice from Equations 2.16 and 2.17 that, unlike instantaneous weight changes, the momentum term is used to change the weights in combination with batch mode learning. In batch mode learning the entire training set is presented to a network once and then the error is calculated. The momentum term, $\alpha$, is set between 0 and 1. If $\alpha = 0$, the current weight changes are not affected by past weight changes at all. If $\alpha = 1$, the weight change is set equal to the last weight change plus the current gradient [14]. While a high value of $\alpha$ reduces the risk of getting stuck in a local minimum, there is an increased risk of overshooting the actual minimum of the error surface. Typically, the momentum term is set at a value of $\alpha = 0.9$ [11,14,22].

*2.3.6 Training the Neural Network.* After the data has been preprocessed, it is used the train the network. The data can be divided into three groups: a training

set, a training-test set, and a validation set. The purpose of the training set is, as its name implies, to train the neural network. That is, to establish appropriate weights for classification on a validation set. The training-test set is used to avoid overfitting of the neural network. Overfitting will be discussed in a later section. The validation set is used to make sure the network is producing expected output. There are many ways to divide the data set into these three groups. One method is to use two-thirds of the data for testing and one-third for validation. Typical divisions have been: the training set - 40%, the training-test set - 30%, and the validation set - 30%. Values of 50/25/25 have also been used.

After the division of the data has been decided, training of the network can begin. Naturally, there is some point when the training of the network will have to stop. To stop training, we limit the number of epochs presented to the neural network. This limit is determined by measuring the error. Error distances (differences between observed output and actual output) are sampled and averaged over fixed interval of epochs. If the average error distance for the most recent fixed interval is not better than (less than) that for the previous fixed interval, the conclusion can be drawn that no progress is being made and training should be stopped [27]. Training on a neural network can also be stopped when the average training error has reached a predetermined target value [22]. The training-test set can also be used to determine when the neural network has been sufficiently trained. Training of the network will begin with the training set. Every so often, the training can pause and the training-test set can be presented to the network to get a measure or error. As long as the error keeps reducing, training should continue. As soon as the error begins to climb, training should stop. When the error begins to climb, overfitting of the network is occurring. Just as it is possible to overfit a model in regression analysis, it is possible to overfit a neural net. Overfitting means that the network has been trained so well on the training set, it doesn't have the flexibility to accurately model other examples, even if they are just a bit different from the

examples presented in the training set. For example, say we are training a network to classify animals into the groups dogs or cats. As a training set we present characteristics of a Dalmation and an alley cat. Say the network was not stopped in time and overfitting has occurred. We have a new set of inputs that we want to try and classify as a dog or a cat. Say this validation set consists of Dobermans and alley cats. Since we overtrained the network we get a 100% classification on the alley cats, however all of the Dobermans are not recognized as dogs. The only dogs the network will recognize are Dalmations. Since we have no predictive power at all, this network must be scrapped. As soon as an increase in error is observed, training should stop and the weights should be set at the values that produced the lowest error on the training-test sample [18].

### *2.4 Saliency Metrics and Saliency Screening Methods*

Not only is architecture of the neural network important, but the quality of the input data is also very important. A parsimonious, salient set of data is desired as input into a neural network. Many times an abundance of data is collected in a study but only some of that data is actually used in the classification process. The other data has little or no effect on the final classification. Not only may there be a lot of data to go through, there is also a potential for a lot of noise to be contained in the data. When dealing with models a general principle always holds: GIGO, "garbage-in, garbage-out." This simply means that we want the best possible data to enter our network. If garbage is put into the neural network, garbage is exactly what will come out. To avoid the "garbage-in, garbage-out" dilemma it is necessary to perform screening on the set of input features. Screening allows the set of input features to be reduced resulting in a parsimonious, salient set of features, as well as cutting down on the network runtime. We discuss three saliency metrics:

- Ruck's saliency metric
- Tarr's saliency metric

- Signal-to-Noise Ratio (SNR) saliency metric

Each saliency metric uses a different method to rank order the set of input features. These metrics will be discussed in turn below.

*2.4.1 Ruck's Saliency Metric.* Ruck's saliency metric sums the network outputs with respect to a given feature using a trained neural network [5, 14, 19, 21]. Ruck's saliency metric is computed as follows:

$$\Lambda_i = \sum_P \sum_M \sum_R \sum_K \left| \frac{\partial z_k^2}{\partial x_i}(x_p^{m(r)}, w) \right| \tag{2.18}$$

where

$P$ = the number of exemplars

$M$ = the number of input features

$R$ = the number of steps that the range of each feature is uniformly divided into

$K$ = the number of network outputs

The derivative in the equation is evaluated at the $p$th input exemplar and trained neural network weights, $w$. Below is Ruck's derivative in detail.

$$\frac{\partial z_k^2}{\partial x_i} = z_k(1 - z_k) \sum_j w_{jk}^2 \delta_j^1 w_{ij}^1 \tag{2.19}$$

where

$\delta_j^1$ = $x_j^1(1 - x_j^1)$ where $x_j^1$ is the output of node j in the hidden layer

$z_k$ = the output of the node k in the output layer

$w_{jk}^2$ = the weight connecting the hidden layer with the output layer

$w_{ij}^1$ = the weight connecting the input layer with the hidden layer

The features are rank ordered according to the average saliency metrics over several training cycles (typically about 30) [5, 14, 19].

*2.4.2 Tarr's Saliency Metric.* Tarr's saliency metric steps away from the derivative and is based solely on the weights. This tactic is taken from the hypothesis that the features that are most important will have bigger weights leading from the input layer to the hidden node layer. The features that are not so important have small weights leading from the input layer to output layer [21]. Tarr's metric is a summation of the squared values of weights connecting the features input node to the hidden layer nodes [5, 14, 19]. The formulation is given as follows:

$$\tau_i = \sum_{j=1}^{J} (w_{ij}^1)^2 \tag{2.20}$$

where

$\tau_i$ = the Tarr saliency metric for input feature $i$

$w_{ij}^1$ = the first layer weight between input node $i$ and hidden node $j$

As with Ruck's metric, features are rank ordered according to their saliency. Large values of $\tau_i$($\Lambda_i$ for Ruck's measure) imply the feature is salient, while small values of $\tau_i$ imply the feature is not salient. The effectiveness of weight-based saliency depends on two things [19]:

1. $w_{ij}^1$ (for all $i, j$) must be from a trained neural network of appropriate complexity.
2. Input features must be normalized to have approximately the same ranges.

*2.4.3 Signal-to-Noise Ratio (SNR) Saliency Metric.* The SNR saliency metric operates on the same general principal as Tarr's metric [?]. That is, the metric relies on the sum of squared weights connecting the input node layer to the

hidden node layer. The signal-to-noise ratio does take a new twist on calculating a saliency measure. The SNR relies on a direct comparison of a feature to an injected noise feature. The calculation for the SNR is as follows:

$$SNR_i = 10 \log \frac{\sum_{j=1}^{J} (w_{ij}^1)^2}{\sum_{j=1}^{J} (w_{Nj}^1)^2} \tag{2.21}$$

where

$SNR_i$ = the saliency metric for the $i$th feature

$J$ = the number of hidden nodes

$w_{Nj}^1$ = the weight connecting the injected noise feature, $x_N$, to the hidden node layer

$w_{ij}^1$ = the weight connecting the input feature, $x_i$, to the hidden node layer

The SNR works in the following fashion. A random noise feature (distributed UNIF(0,1)) is added to the existing input vector. The weights connected to the noise input node should be relatively small because the added noise contributes nothing to the overall process being evaluated. On the otherhand if the feature is salient, then its weights are relatively large. So in the case of a salient feature the SNR is a large number on top of a small number. The resulting ratio is significantly larger than zero, which indicates the feature's saliency. In contrast, nonsalient features create a ratio close to one, indicating a nonsalient feature. Like Ruck's and Tarr's metrics, the SNR ranks orders the features based on saliency value.

Having assigned saliency values to the input features, screening methods are applied to sort through the features and decide which to keep and which to toss. There are three different screening methods discussed in the following sections. They are:

- Belue-Bauer [5]
- Steepe-Bauer [19]

- Signal-to-Noise screening method [4]

*2.4.4 Belue-Bauer Screening Method.* The Belue-Bauer screening method makes use of an injected noise feature to distinguish between salient features and nonsalient features [5, 21]. The following is the procedure used to determine the significant feature inputs:

1. Introduce noise feature to the original set of features.
2. Train the neural network.
3. Compute the saliency of all features (use Ruck's or Tarr's saliency metrics).
4. Repeat steps 2 and 3 at least 30 times (with weights being randomly initialized and training and test sets being randomly selected at the beginning of each training cycle).
5. Assume the average saliency of the noise feature is normally distributed. Find upper one-sided ($\alpha \times 100$) percent confidence interval for the mean value of the saliency of noise.
6. Choose only those features whose average saliency values falls outside of this confidence interval.
7. Retrain the network with salient features.

Salient features will have means significantly different from the noise feature (it will not fall within the confidence interval.) The noise feature will be close to zero while the salient feature will not. Ruck's or Tarr's metric can be used for this screening method. Even though they measure saliency differently, the outputs are pretty much the same when the Belue-Bauer screening method is applied.

*2.4.5 Steepe-Bauer Screening Method.* The Steepe-Bauer screening method applies a Bonferroni approach to calculating the statistical significance of a feature [19, 21]. The Bonferroni approach considers a hypothesis on a 'family' of tests. In

this case, the family of tests looks for a difference in the means of the actual feature's saliency and the injected noise feature's saliency. The Bonferroni approach is applied to $M$ individual hypothesis tests to achieve a predetermined 'family' significance level of $\alpha$. The Bonferroni critical value is defined as a t-statistic as follows:

$$B = t_{\frac{\alpha}{M},\upsilon}$$

where $\upsilon$ = N -1, $t$ = the t-statistic with $\upsilon$ degrees of freedom, and N = number of neural networks.

The following is the procedure for the Steepe-Bauer screening method:

1. Augment feature set with noise feature, $x_N$.
2. Use the augmented set to train N neural networks.
3. For each candidate feature, test whether the candidate feature's average saliency is different that the noise feature's average saliency.
   - Compute test statistic t*. This statistic is based on the difference between the two feature saliencies. More can be found in Steppe's feature screening article [19].
   - Evaluate the test statistic using the Bonferroni critical value, **B**.
     i. if t* $\leq$ **B** feature $i$ considered nonsalient
     ii. if t* $>$ **B** feature $i$ considered salient
4. Eliminate nonsalient features.
5. Retrain neural network using only the salient features.

Step 2 mentions training N networks. It has been found that N=10 is sufficient. Slightly modified versions of Ruck's and Tarr's saliency metrics were used to assign feature saliency.

*2.4.6 Signal-to-Noise Ratio Screening Method.* The SNR screening method is less statistically rigorous than the previously mentioned methods [14,21], however, it does have some important advantages. The SNR screening method follows these steps:

1. Add a Uniform(0,1) noise feature, $x_N$, to the original feature set.
2. Standardize all features to zero mean and unit variance.
3. Randomly initialize the weights between -0.001 and 0.001.
4. Randomly select the training and test sets.
5. Begin training neural network.
6. After each epoch, compute the SNR saliency measure for each input feature.
7. Interrupt training after saliency metric values stabilize.
8. Compute the test set classification error.
9. Identify the feature with the lowest SNR value and remove it from further training.
10. Continue training the neural network.
11. Repeat steps 6 - 9 until all of the features in the original set have been removed.
12. Compare reaction of the test set classification error rate to the removal of the individual features.
13. Retain the first feature whose removal caused a significant increase in the test set classification error rates as well as all features that were removed after that first salient feature.
14. Retrain the neural network with only the parsimonious set of saliency input features.

There are many advantages to the SNR screening method and SNR metric. The SNR screening method is a quick, rough initial screening of the input features.

This quick estimate gets rid of any apparent noise. The SNR is especially useful when time is of the issue and it is more important to rapidly screen out unwanted features than to get the best possible feature set. The SNR screening method is also interruptible. It is designed such that the user can stop in the middle of the process and remove unwanted features. Once again, this cuts down on the processing time and gives an initial rough estimate. The Belue-Bauer [5] and Steepe-Bauer [19] screening methods provide a finer tuned screening. After the SNR screening method is applied, the Belue-Bauer or Steepe-Bauer screening method can be used to screen out any borderline or questionable features.

This research effort uses the SNR metric and screening method. While statistical methods are not the backbone of this screening method, the SNR metric has been found to be fairly robust as shown by Sumrell [21]. Factors considered included the number of hidden nodes, learning rate, and momentum rate. Sumrell found that the SNR metric is fairly robust across all network architectures. The number of hidden layer nodes and changes in the learning rate had marginal changes in classification accuracy. However, it was also noticed that high momentum rates resulted in poor classification. The recommendations passed on by Sumrell include using N to 3N hidden nodes (N is the number of input features), a learning rate between 0.1 and 0.9, and a momentum rate between 0.1 and 0.5 [14].

### *2.5 Multivariate Discriminant Analysis*

Discriminant analysis is a "technique for classifying individuals or objects into mutually exclusive and exhaustive groups based on an observed set of independent variables [3]." The goal behind discriminant analysis is to attach a scalar score to each object. This section discusses how that score is calculated and the conditions surrounding that calculation. Then we investigate each variable's contribution to forming the discriminant score. Finally, we look at estimating the error rate associated with a discriminant function.

*2.5.1 Discriminant Score.* As mentioned above, it is desired to get a scalar score for each object in the data set. This score will determine to which group the object belongs. This score ideally holds certain attributes:

- a linear combination of the object's attributes
- the mean of the two groups are as far apart as possible
- small variance

Performing discriminant analysis can be summed up in a few simple steps:

1. Check for multivariate normality. This is a must for discriminant analysis.
2. Check to see if $\Sigma_1 = \Sigma_2$, where $\Sigma_i$ is the covariance structure for group $i$. This is not a hard requirement to perform discriminant analysis as there are discriminant methods that can be utilized if $\Sigma_1 \neq \Sigma_2$.
3. Compute the discriminant function.
4. Validate the discriminant function.

Several discriminant methods can be used to form the discriminant function. The discriminant function used in this research effort is the quadratic discriminant function. The quadratic discriminant function has several advantages that make it favorable over other methods. One advantage to the quadratic discriminant function is that it is does not require $\Sigma_1 = \Sigma_2$. Another advantage to the quadratic discriminant function is that it allows classification if the different groups are not linearly separable. An example of this is the XOR problem, shown earlier in Figure 2.2. The formula for calculating the quadratic discriminant scores is,

$$d_i^Q(\underline{X}) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\underline{X} - \widetilde{\mu}_i)^T\Sigma_i^{-1}(\underline{X} - \widetilde{\mu}_i) + \ln P_i \tag{2.22}$$

where

$$
\begin{aligned}
d_i^Q &= \text{the quadratic discrimiant score for group } i \\
\Sigma_i &= \text{the covariance structure for group } i \\
\underline{X} &= \text{the new exemplar} \\
\widetilde{\mu}_i &= \text{the estimation of mean for group } i \\
P_i &= \text{the prior probability of belonging to group } i
\end{aligned}
$$

If $P_i$ is equal for each group then the $\ln P_i$ can be dropped from Equation 2.22. This situation occurs when the number of samples from each group are equal. Likewise, if the covariance structures for each group are equal, the pooled covariance structure is used in place of the individual covariance matrices. The pooled structure can be computed as,

$$
S = \frac{1}{N_1 + N_2 - 2}(\underline{X}_{d(1)}'\underline{X}_{d(1)} + \underline{X}_{d(2)}'\underline{X}_{d(2)}) \tag{2.23}
$$

where

$$
\begin{aligned}
S &= \text{the pooled covariance estimate} \\
N_1 &= \text{the sample size from group 1} \\
N_2 &= \text{the sample size from group 2} \\
\underline{X}_{d(i)} &= \text{the centered data matrix from group } i
\end{aligned}
$$

Once the quadratic discriminant score, $d_i^Q$, has been computed for all $i$, the exemplar is classified into the group that has the highest $d^Q$ score. As with artificial neural networks, creating a salient group of input features for discriminant classification can lead to a better probability of correct classification. The next section discusses ways to assess which variables are important for classification.

In addition to the quadratic score discussed above, a linear score can be obtained from Equation 2.22 with a few minor modifications. The linear classifier assumes that the covariance matrices for each group are statistically equal ($\Sigma_1 = \Sigma_2$). The equality of the covariance matrices leads to a few terms being dropped out of the quadratic score. The final linear classifier looks like,

$$\begin{aligned} d_i^l(\overline{X}) &= \widetilde{\mu}_i^T \Sigma^{-1} \overline{X} + w_i \\ w_i &= -\frac{1}{2} \widetilde{\mu}_i^T S^{-1} \widetilde{\mu}_i + \ln(P_i) \end{aligned} \tag{2.24}$$

where

$$\begin{aligned} d_i^l &= \text{the linear discriminant score for group } i \\ \widetilde{\mu}_i &= \text{the estimator of mean for group } i \\ S &= \text{the estimated pooled covariance matrix} \\ P_i &= \text{the prior probability of belonging to group } i \end{aligned}$$

The principle for classifying new exemplars using the quadratic score also applies to the linear score. A new exemplar receives a linear score for each group. The highest $d^l$ score for the new exemplar indicates which group that observation is classified into. Further discussion on discriminant analysis to include the computation of discriminant scores can be found in Bishop, 1995 [6].

*2.5.2 Variable Contribution.* As previously mentioned, a higher probability of correct classification can be obtained if redundant variables are removed from formation of the discriminant score. In order to remove variables, variables significant to classification must be determined. Traditional approaches to determine variable contribution use group means and univariate F-values for each variable and/or magnitudes of standardized discriminant weights. Although traditional, there are problems with these methods. If the variables are intercorrelated, the conclusions

obtained by these traditional methods can be misleading [10]. For example, if standardized discriminant weights are found and the variables are intercorrelated, the discriminant weight will be split between the two variables. This can make both variables seem to contribute marginally to the discriminant function when in reality only one is important. F-values can also be inaccurate because interdependence is ignored. Another way to determine variable contribution is to use discriminant loadings. Discriminant loadings are defined as "correlation of a variable with the discriminant function [10]." The loadings can give an accurate feel for which variables are actually important to that function. The discriminant loadings can handle variable intercorrelation better than traditional methods. They are also easier to interpret than standardized discriminant weights. One final method to determine variable contribution is to use partial F-values. Recall that tests based on univariate F-values are unable to adequately handle intercorrelated variables. Partial F-values partition out the variance of a variable that is already explained by other variables [10]. In other words, it takes the intercorrelation out of consideration and just reports the variation due to the variable of interest.

Out of the three methods mentioned above, Dillon and Goldstein [10] prefer the discriminant loading method. It gives a simple interpretation of how important each variable is to the discriminant function without being affected by variable intercorrelation. However, there is one inherent problem with the method of discriminant loadings. The calculation of the loadings require calculation of discriminant weights. These discriminant weights can only be calculated if the covariance structure for each group are statistically equal, meaning the pooled covariance structure can be used (see Equation 2.23). If it is shown that the covariance structure of the groups are not equal, discriminant loadings cannot be used.

*2.5.3 Error Rate Estimation of Multivariate Discriminant Classifiers.* After the proper variables have been screened out and a discriminant function is formed, an estimate of the error rate can be obtained. This estimate gives insight into the

ability of the discriminant function to classify data. There are several ways of obtaining an estimate for the error rate. Three methods are discussed here. These methods are: resubstitution, data splitting, and Lachenbruch's holdout procedure.

The resubstitution estimate of error rate for a discriminant function is a simple calculation. The error rate is simply the proportion of misclassified observations, using all the original data. This method can be most easily explained by examining a confusion matrix, shown below.

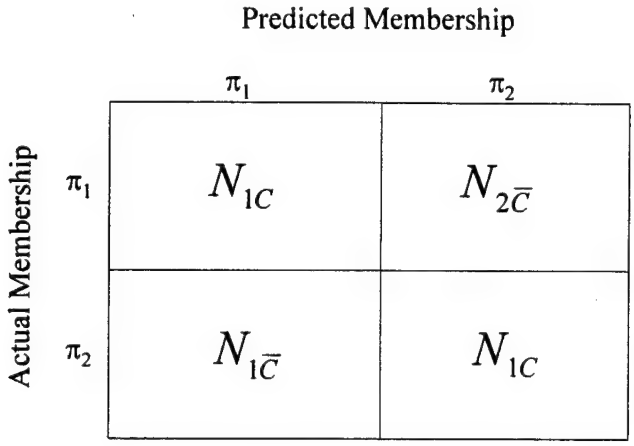


Figure 2.5 Confusion Matrix

$\pi_1$ = group 1

$\pi_2$ = group 2

$N_{1C}$ = the number in group 1 classified as group 1

$N_{2\overline{C}}$ = the number in group 1 classified in group 2

$N_{1\overline{C}}$ = the number in group 2 classified in group 1

$N_{2C}$ = the number in group 2 classified as group 2

$n_1$ = the total number in group 1

$n_2$ = the total number in group 2

The estimate of the error rate is most commonly known as the apparent error rate (APER). The APER can be calculated as follows:

$$APER = \frac{N_{1\overline{C}} + N_{2\overline{C}}}{n_1 + n_2} \tag{2.25}$$

One failing of the resubstitution method is that it tends to underestimate the actual error rate.

The second method for error rate calculation is data splitting. With this method, the total data set is split into two sub-sets. The first sub set (usually $\frac{2}{3}$ of the total data set) is used to construct the classification rule. The remaining data (the $\frac{1}{3}$ that is left) is used to validate that discriminant function. This validation can also be done using a confusion matrix and the APER. This APER obtained using the validation set will be a bit more accurate than the APER obtained using the resubstitution method. One variation of this method is to randomly split the data multiple times. An APER can be obtained for every validation set and an average APER is computed. This gives a better feel for the true error rate associated with the discriminant function. These estimates are consistent and unbiased. However, a large data sample is required to accomplish multiple splits of the data [10].

The last estimation of the error rate can be obtained using Lachenbruch's holdout procedure. In this procedure all but one observation from the total data set is used to form the discriminant function. The observation held out is then passed through the function and assigned to a group. This procedure is repeated for all $m$ points in the data set. After all data points have been classified, an estimate of the expected actual error rate can be obtained. The equation for the expected actual error rate is,

$$E(AER) = \frac{N_{1m}^{(H)} + N_{2m}^{(H)}}{N_1 + N_2} \tag{2.26}$$

where

$$
\begin{aligned}
N_{im}^{(H)} &= \text{the number of misclassifications } m \text{ of objects of group } i \\
N_1 &= \text{the number in group 1} \\
N_2 &= \text{the number in group 2}
\end{aligned}
$$

This holdout procedure yields a nearly unbiased estimate of the misclassification probabilities [10], but can be computationally intense if the data set is very large. Dillon and Goldstien [10] state that data splitting and the Lachenbruch holdout procedure are most reliable for estimating error rates associated with a discriminant function.

### *2.6 Psychophysiological Features*

A tremendous amount of research has been done investigating the effects of mental workload on physiological features. Measures of these physiological responses have been associated with psychological states, thus the term, psychophysiological features: psycho meaning "mental activities or processes [24]" and physiological meaning "all the functions of a living organism and their parts [23]." Several recent research efforts have concentrated on analysis of psychophysiological responses in multi-task environments [9, 13, 28–32]. Psychophysiological methods have several advantages when studying multiple task environments. These advantages include: the measures are continuous (they can be collected throughout the study); the collection of the features does not inhibit the subject from completing the primary task (they are non-inhibitive); the features are relatively robust; and the features are easy to collect [28]. From the variety of physiological features that can be monitored, the following measures are often collected:

- Cardiac Measures
- Respiratory Measures

- Hormone Measures
- Ocular Measures
- Brain Activity Measures

*2.6.1 Cardiac Measures.* Measures of the heart have been used in multi-task environments as early as 1932 when heart rate was used to measure pilot responses during flight [9]. Since then, several studies have been conducted measuring heart rate in the flight environment [13,28,29,32]. Heart rate has been shown to be sensitive to several variables including, landing at different airports, refueling during transatlantic helicopter flights, using autopilot to land aircraft, simulated instrument landings, pilot versus copilot flying the aircraft, combat missions, and surface attack training missions [9]. In general, heart rate increases as cognitive workload increases. Typical high workload flight segments are takeoffs, landing, touch and go, etc. [13,28,29,32]. A second cardiac measure that can be obtained is the heart rate variation (HRV). The heart rate variability is the variation of the heart rhythm. In general, HRV is thought to decrease as mental workload increases. However, there is a lot of controversy surrounding the use of HRV as a viable psychophysiological feature.

One problem encountered in using HRV is the question of how to measure HRV. Multiple measures of HRV are available; perhaps as many as 26 measures [28]. It is unclear if some measures are better than others, if there is one best measure, or if certain measures should be used in certain situations. One widely used measure of HRV is a spectral analysis. It is thought that spectral analysis may be useful in determining mental workload, however, this fact has yet to be proven for the multi-task environment [9]. Another problem that has been encountered in measuring HRV is the different results that have come out of studies. Some studies show that there is a definite advantage to collecting HRV, others suggest that there is no advantage to collecting HRV. For example, a study (summarized in Damos)

was conducted on underwater diving while performing a task [9]. Two sets of subjects were used, inexperienced divers and experienced divers. The inexperienced divers showed that HRV did indeed decrease while performing the task in the water. However, the experienced divers showed no change in HRV. Another study reported in Damos suggested that HRV may decrease due to aging, making HRV questionable after a certain age. The biggest controversy surrounding HRV is whether or not it adds anything to the study beyond considering just heart rate, especially in the multi-task environment. Many studies suggest that heart rate may be more sensitive to changes in cognitive workload, implying that heart rate alone may be an adequate measure [9, 28, 29].

*2.6.2 Respiratory Measures.* Few studies have been conducted on the use of respiration as a measure for cognitive workload. Those that have been conducted suggest that respiration rate increases as workload increases [9, 28, 32]. Respiration measures are typically collected using bands that strap around the chest of the subject. Tasks that involve voice communications can create a potential problem in measuring respiration because speech disrupts the pattern of breathing [9]. It has been suggested that voice analysis be used as a measure of cognitive workload. Fatigue and stress due to increased workload are thought to cause measurable changes in voice pattern [9].

*2.6.3 Hormone Measures.* A few studies have been done measuring the hormone levels in a subject [9]. A high mental workload indicates that a very stressful situation has been presented to a subject. In response to stress, the sympathetic nervous system is stimulated. This stimulation causes the adrenal glands to release hormones into the blood stream. These hormone levels can then be collected after the task has been completed and a determination can be made on the workload that was experienced by the subject during the task. Hormone levels can be collected via blood, urine, or saliva samples. Even though hormone measures are very definitive

when it comes to determining mental workload, there is a major drawback. Hormone levels cannot be collected until after the task has been completed. Collection of hormone levels after a multi-task study can be almost a moot point. Such tasks usually contain several levels of mental workload from low to overload. It could be difficult or near impossible to try and correlate the hormone levels with specific events. Because of these limitations, hormone levels are not typically collected and used in trying to classify mental workload in certain multi-task environments such as flight.

*2.6.4 Ocular Measures.* Measures of eye blinks have been collected in simulations as well as in actual flight environments [9,13,31,32]. In general, it has been found that as visual demands increase both blink rate and blink duration decrease. Blink duration is defined as the time spent blinking. The subject can process more information and will not miss information if the blinks are less frequent and very fast. There are a few problems that can arise when measuring eye blinks. First of all, eye blink measures may be very good measures of mental workload when examining tasks that involve processing visual information. They may not be as informative when the tasks involve cognitive workload [9]. Curious results have also been reported when examining flight versus ground segments of a task [9,32]. It was shown that there were higher blink rates during the flight segments. This was initially thought to contradict the theory that as cognitive workload increases blink rate decreases. However, this apparent contradiction is thought to be due to the increased visual information that is inherent with flight. When blink rates were examined in the flight only, the trend of decreased blink rate during increased workload levels did indeed hold [9]. Another result that has been observed in studies is the sensitivity of blink rate versus blink duration in high workload environments. It looks as though blink duration is more dependent on the amount of visual information that is being presented to the subject, regardless of the actual cognitive workload that is being

presented [32]. Blink rate may actually be more sensitive to the actual cognitive workload level than blink rate [9].

*2.6.5 Brain Activity Measures.* Perhaps one of the most significant psychophysiological features is the brain electrical activity. The brain has constant electrical signals running across it. By placing electrodes on the scalp, measures of these electric impulses can be recorded. A continuous plot of these impulses can be created. This plot of microvolt changes over time is known as an electroencephalograph (EEG). The EEG has been used in multiple studies in the multi-task environment involving mental workload levels [9, 11, 13, 14, 30, 32]. "EEG normally includes a composite of waveforms that demonstrate a frequency range of 1 to 40 Hz [9]." When used for evaluating mental workload status, frequency ranges of 1 to 40 Hz are typically considered. According to Jared Lambert, AFRL/HE, frequencies below 1 Hz are usually associated with eye blinks and frequencies above 40 Hz are attributed to muscle movement. As mentioned before the EEG is a continuous composite of waveforms. All of the frequencies above are squashed into one wave. The range of 1 - 40 Hz can be separated into 5 power bands of frequencies that can be measured via EEG. Table 2.1 gives a breakdown of these distinct power bands.

Table 2.1 Frequency Band Designations.

| **Band** | **Symbol** | **Frequency** |
|---|---|---|
| Delta | $\Delta$ | 1-3 Hz |
| Theta | $\theta$ | 4-7 Hz |
| Alpha | $\alpha$ | 8-12 Hz |
| Beta | $\beta$ | 13-30 Hz |
| UltraBeta | $\mu\beta$ | 31-42 Hz |

Table 2.1 represents the power at each frequency typically included in analysis. However, the raw EEG must be transformed from its initial composite waveform to these individual bands. When considering EEG measures, the continuous EEG measure is typically broken down into segments such that the average amplitude of power for each given band can be determined [12,15]. Fourier transforms are used for

conversion of sinusoidal, time-domain waveforms into frequency-domain waveforms. Any continuous wave can be written as a linear combination of sinusoidal waves, therefore our composite wave, time series EEG can be described in terms of the frequency components of the signal [12, 15]. The classical mathematical approach to Fourier analysis can be very frustrating for all but the simplest waveforms. The calculations can get very intensive. Interpolation of $2m$ data points using classical methods requires $(2m)^2$ multiplications and $(2m)^2$ additions. If the waveform contains thousands of data points, the calculations can run into the millions. As an alternative, complex waveforms can be sampled and digitized with a waveform digitizer. When the waves are digitized, FFTs (Fast-Fourier Transforms) can be used to evaluate the wave. In 1965, J.W. Cooley and J.W. Tukey described the FFT algorithm. Compared to the brute force classical method of calculation, the FFT requires on the order of $(m \log_2 m)$ multiplications and $(m \log_2 m)$ additions. If the waveform contains thousands of points, the calculations will stay in the thousands. Note that $m$ is a power of 2 ($m = 2^k$). Alternative methods for Fourier analysis are done on numbers of data points that are not a power of 2. This is a considerable improvement over the millions of calculations done with the classical method of calculation [7]. Fourier transforms of the EEG data provide the power bands that are typically used in the analysis of mental workload in multi-task environments.

Generally, the alpha ($\alpha$) and theta ($\theta$) bands have been most useful in the measuring of mental workload. Alpha band activity has been found to decrease with increased cognitive demands while theta band activity tends to increase during increased cognitive demands [9, 13]. On the other hand, during low workload levels, alpha band activity is shown to increase while both theta and beta ($\beta$) band activity decrease [9].

*2.6.6 Summary of Psychophysiological Features.* Overall it has been shown that the use of multiple measures of psychophysiological features give a greater insight into the mental workload level of a subject over the use of separate mea-

sures [9, 29, 31]. Among the psychophysiological features that have been discussed, cardiac measures, respiratory measures, ocular measures, and brain activity measures seem to be the most readily available and least intrusive for measuring workload level in multi-task environments. Once again, the hormone measure may not be practical in the multi-task arena. Because of the different stresses created on a subject in multi- task environments, the use of multiple measures should give a more complete picture of the actual cognitive load of the subject. One important fact remains to be discussed when considering psychophysiological features for study. This is the difference in laboratory studies and actual studies. It has been discovered that while laboratory data provides useful observations and theories, it is better to collect real world data to analyze real world scenarios. Comparisons between laboratory data and real world data has shown that while they produce similar effects on some physiological variables, they can produce different effects on others [29]. These differences support the notion to use real world measures to analyze real world scenarios rather than trying to extrapolate real world answers from the laboratory data [28]. This difference in laboratory data and actual real world data also supports the notions that several physiological measures should be used when evaluating complex situations [29].

# *III. Data Collection and Preprocessing*

This chapter discusses the experiment and data collected by AFRL/FPL. The first section explains the conditions and purpose of the experiment. The second section discusses the raw physiological features collected during the experiment. This chapter also examens the preprocessing that was required on all the EEG, heart, eye and respiration data. The fourth section summarizes what the final input data matrix looks like, whether it be the input matrix into a neural network or a statistical classifier. Finally, an investigation is presented that details some of the properties of the data, to include the input feature correlations and the potential for outliers.

## *3.1 The Experiment*

The experiment conducted by AFRL/HE was an actual flight experiment. The Wright-Patterson Aero Club had ten volunteers step forward as subjects for the experiment. The research lab created a predetermined flight route containing varying workload levels. Each of the ten volunteers flew the same flight route two times, on different days. The flight segment itself was divided into 22 two-minute segments. A technician from AFRL flew with the pilots, monitoring data collection and transitions between workload levels. In addition to the pilot and the lab technician, a copilot was present for safety reasons, but was not part of the experiment. Each two-minute segment within the flight has a certain workload level associated with it. Appendix A lists the flight segments and their associated workload levels.

The flight route was designed to contain three distinct workload levels: low, medium and high. The lab determined what difficulty level to associate with each flight segment. In addition to receiving the lab's input on the workload level associated with each flight segment, each pilot's subjective measure of the workload level associated with each flight segment was also provided. Appendix B shows the pilot's subjective measures of workload level associated with each flight segment.

There were some discrepancies between workload levels according to the lab versus those according to the pilot. For example, the lab determined the VFR touch-and-go portion of the flight to be a high workload level, and the IFR airwork portion of the flight as a medium workload level. However, the pilots rated both the IFR airwork and VFR touch-and-go segments as high workload. A compromise had to be reached between what the lab thought was hard and what the pilots thought was hard. The touch-and-go segment of the flight was thought to be hard by the pilots and by the lab. Using this point as the minimum high workload level for all high workload levels, a line was drawn through the VFR touch-and-go segment across the page, as seen in Figure B.1 in Appendix B.. For the purposes of this research, all flight segments below that line are considered medium and low workload levels. Everything above the line is considered a high workload level.

Transitions between flight segments raise another concern about measuring workload levels. Transitions between workload levels are not instantaneous. The pilot doesn't go from cruise to a touch-and-go instantaneously. It is possible that the actual workload level will transition in the middle of a flight segment. What this means is, the pilot could be flying in the cruise segment, and the physiological readings will begin to register a change in mental workload level before he actually gets to the touch-and-go segment. For the purposes of this thesis effort, all transitions are considered to be instantaneous, although it is realized that classification error could be caused by the uncertainty of the transitions between flight segments.

### *3.2 Data Collected*

Several different physiological features were collected for this experiment including electroencephalography (EEG) electrode readings. The pilot was required to wear a special cap fitted with 29 electrodes. Figure 3.1 shows a diagram of the head fitted with the numerous electrodes.

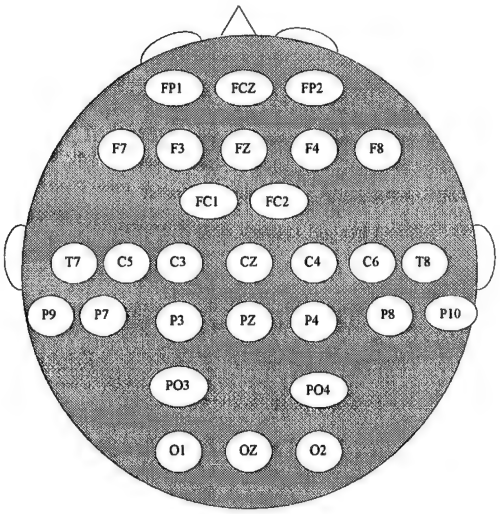


Figure 3.1 Electrode Placement

Each electrode has a specific name associated with it. The location and naming of the electrode sites are based on the International 10-20 system [14]. The EEG locations are labeled with a letter (sometimes two) followed by a number. The letter designates the brain region, while the associated number indicates the placement of the electrode on the left or right side of the brain. If the number is even, the electrode is on the right side of the brain; odd numbers indicate the left side. The bigger the number, odd or even, the further away the electrode is from the center of the brain, center meaning font nose to back.. The middle has no numerical designator. The letter "Z" indicates the middle of the brain. The following table lists the meaning of the letters associated with each electrode.

Table 3.1 EEG Identifiers

| **Letter** | **Location** |
|---|---|
| C | Central |
| F | Frontal |
| O | Occipital |
| P | Parietal |
| T | Temporal |

After the pilot was fitted with the electrode cap, raw EEG data was collected and sent through a program called *Manscan 4.0.* This program filters out undesirable artifacts from the signal. Examples of undesirable artifacts include eye movement and muscle movements caused by the pilot's head moving around during flight.

In addition to the raw EEG data collected, eye, respiration, and heart data was also collected and assembled in electronic files. These files report the elapsed time in milliseconds between event. An event is a heart beat, eye blink, or a breath taken. In addition to the elapsed time between events, other factors are collected with each physiological feature. The respiration data also includes minimum and maximum amplitudes associated with each breath, the eye data includes the amplitude and the duration of each eye blink, and the heart data only includes the time between heart beats in milliseconds.

### *3.3 EEG Processing*

The goal is to create one set of features that can be used as inputs into either a statistical classifier or an artificial neural network. Before this could be accomplished, a certain amount of preprocessing had to be done to make the code "usable"

Recall that there are 29 electrode sites. The data file provided was raw EEG data collected in two minute segments. An example of the raw EEG data is shown in Figure 3.2.

In the raw data file, two extraneous readings are also collected. These readings are Horizontal Electro-oculography (HEOG) and Vertical Electro-oculography (VEOG). HEOG and VEOG are readings on horizontal and vertical eye movements. These readings were collected in order to take out artifacts in the data due to eye movement. Since these are not EEG readings, they are simply deleted from consideration.

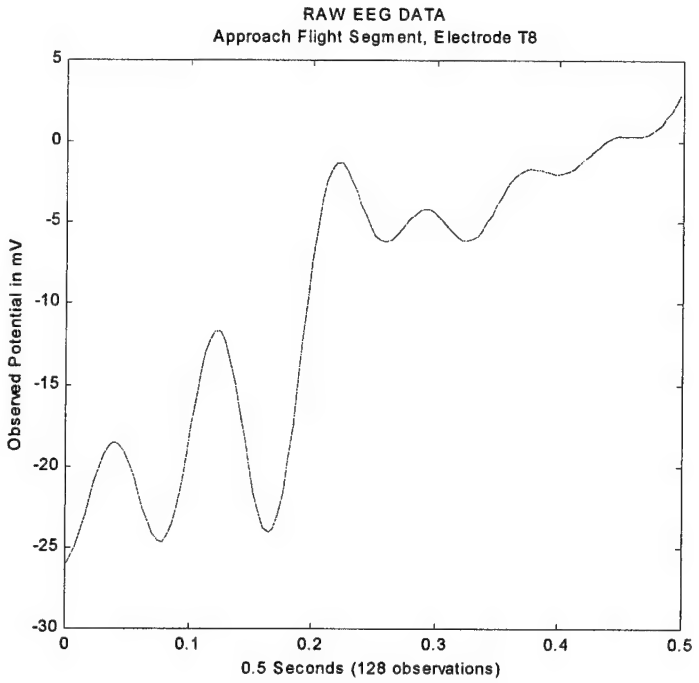


Figure 3.2 Raw EEG Signal from Electrode T8 during the Approach Flight Segment

The raw data that is collected has a time dependency associated with it. In order to use the EEG data as features into a classifier, time dependency needs to be removed from the data. This can be accomplished by passing the raw data through a Fast Fourier Transform (FFT). The FFT takes the data from a time domain to a frequency domain. Transforming into a frequency based domain will enable estimates of power to be obtained. An FFT was performed on each EEG signal for every one second of raw data. According to the Nyquist sampling theorem, estimates for power can only be made for frequencies up to $f_s/2$, where $f_s$ is the sampling frequency [16]. The data provided was collected at a sampling frequency of 256 Hz. Thus, according to the Nyquist theorem, estimates for power can be made up to 128 Hz. *Matlab* code was written to perform 1 second FFTs on all raw EEG data. This produces power estimates from 1 to 128 Hz. An example of power estimates by frequency band over a one second window is shown in Figure 3.3, which

is known as a periodogram. Recall that the frequency bands shown on the diagram were listed in the previous chapter.

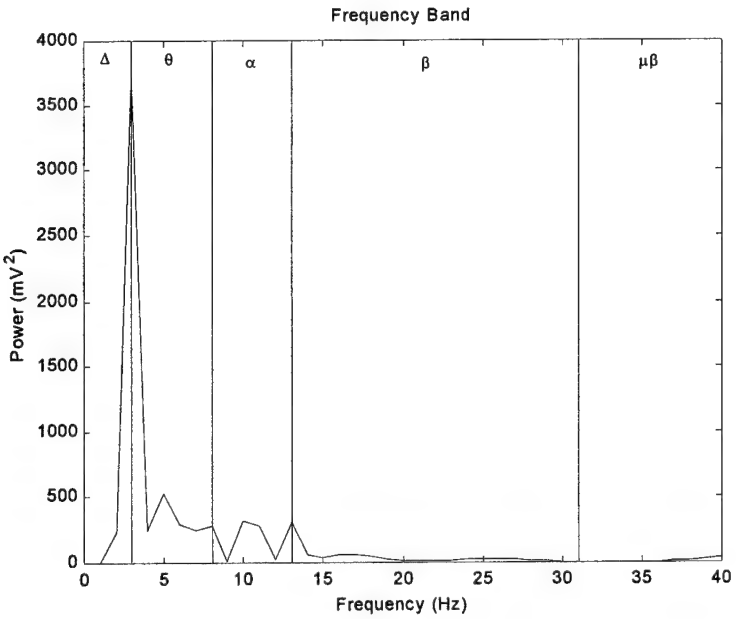


Figure 3.3 FFT at One Electrode for One Second

For the purposes of this research, frequencies from 1 - 40 Hz will be used. The x - axis of Figure 3.3 is frequency in Hz. The vertical lines separate each frequency band, as discussed in the last chapter. The y-axis of the figure represents power, expressed in $\mu V^2$ (microvolts$^2$). The power estimates were calculated in the following manner: an FFT was performed on the raw EEG data over one second intervals. Then the absolute value of the transformed data was squared, giving a power estimate for that one second of data. The final power estimates that were kept were from 1 - 128 Hz, because of the Nyquist theorem.

The periodogram provides a visual picture of the estimate of the power contained in the signal. As with any estimation technique, there is a certain amount of error associated with the estimate obtained. The periodogram estimate of power (either looking at it visually or mathematically) has a large amount of variance as-

sociated with it. Unfortunately, the amount of variance does not decrease as the number of sample sizes increase [16]. In other words, if the FFT of the signal is taken more frequently, say once every half a second, the variance of the power estimate is not reduced. The variance in the power estimates can be reduced by breaking the signal into sections (take one second FFTs) and averaging the power in these separate sections. The more sections that are averaged, the lower the variance in the resulting power estimate [16]. The length of the signal (the frequency at which it was sampled) limits the number of separate sections the signal can be divided into. As a result, overlapping signals can be added to increase the number of sections. The overlapping sections are statistically dependent, resulting in some higher variance. The number of sections settled on depends on how much variance the researcher is comfortable with.

In order to decrease the amount of variance in the power estimate, it was decided that the signal would be broken into one-second sections. Then the power estimate obtained would be averaged over ten-second windows. In order to obtain some amount of further variance reduction, overlapping windows were also included in the analysis. Recall that although these sections are statistically dependent, resulting in higher variance, the more sections the signal is separated into, the lower the variance. Ten seconds of data was averaged, then five seconds were skipped over and the next ten seconds were averaged. This is shown graphically Figure 3.4. Thus, in this research, each two minute window will initially have 120 one-second power estimates. These power estimates are averaged with 12 non-overlapping 10-second windows and 11 overlapping 10 second windows. Therefore, the net result is a total of 23 exemplars of averaged power for each two-minute segment; for a total flight this comes to 506 exemplars (22 two-minute segments).

After acquiring the total power for one second of data, the power for each of the five frequency bands must be collected. In essence, a filter is created to remove only the power in the frequency bands that are relevant to this research effort. The power

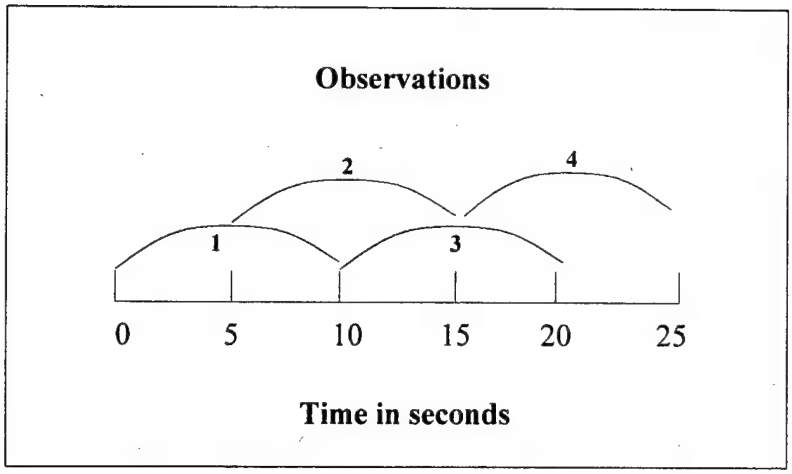


Figure 3.4 Power Estimate Windows

estimates for each frequency band are obtained just by summing all power estimates within the given range of frequencies for that particular band. For example, if a power estimate is collected for the delta band, all power readings given by the FFT between 1 and 4 Hz are summed together giving a total power reading for the delta band for that one second of data. The power is then averaged over 10 seconds with the overlapping windows figured in. The final bit of processing is to transform the data using the $\log_{10}$ of the averaged power for each 10-second window. An example of a fully processed two-minute block of data is shown in Figure 3.5.

The y-axis is $\log_{10}$ of average power in microvolts$^2$ ($\mu V^2$), for each bandwidth, and the x-axis represents seconds. After fully preprocessing the raw EEG data, the end result is five bandwidths at 29 electrode locations resulting in 145 different EEG variables. These variables are labeled according to electrode and bandwidth. For example, the first variable would be electrode C3, delta bandwidth. A summary of the steps needed to preprocess the raw EEG data is shown in the flow diagram, Figure 3.6.

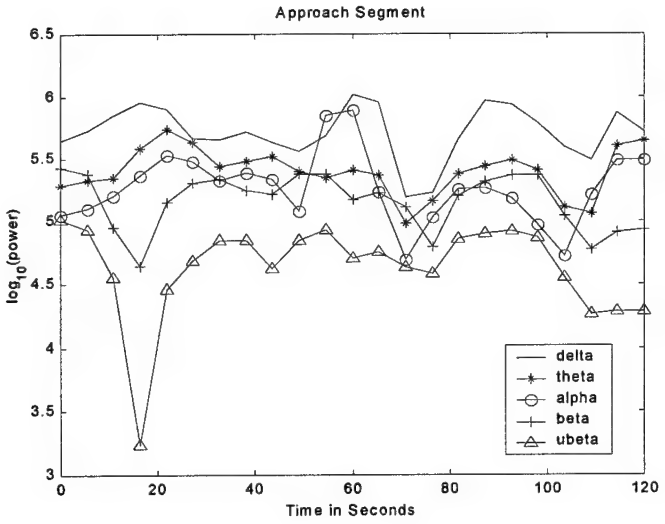


Figure 3.5 Processed EEG Signal Containing 5 Seconds of Overlap

## *3.4 Physiological Feature Processing*

This section discusses the preprocessing performed on the heart eye and respiration files, resulting in six distinct additional features used for classification of mental workload level.

*3.4.1 Cardiac Measures.* The raw heart files contain heartbeat intervals. This is the time (in milliseconds) in between heartbeats for each two minute segment. Preprocessing yields two distinct physiological heart features: heart rate (in beats per minute) and heart rate variability. Recall that heart rate variability can be thought of as how often the heart beats. In order to create exemplars with all physiological features considered over the same time period, the heart features are calculated over the same 10 second windows as the raw EEG data. *Matlab* code was written to create 23 overlapping 10 second windows for average heart rate and heart rate variability. Recall that processing the raw EEG revealed that 23 overlapping windows are created per two minute flight segment.

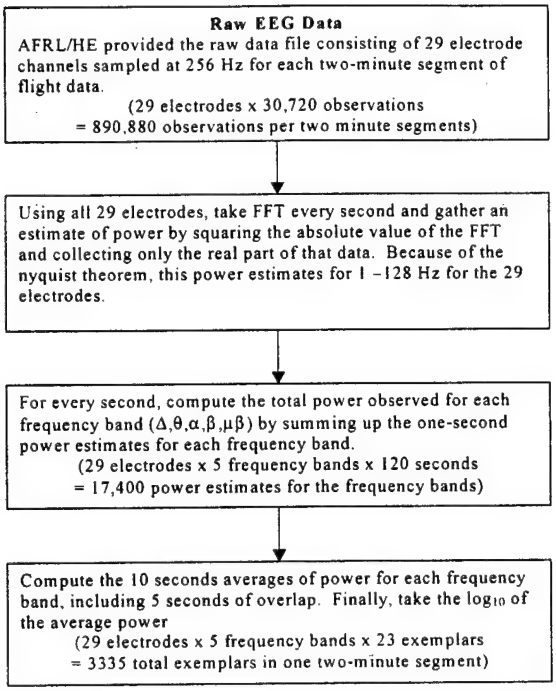


Figure 3.6 Raw EEG Data Processing

First, we consider the variable, average heart rate. *Matlab* code identifies all observed beats within a given 10 second window and calculates the average interval between beats over that 10 second window. This interval is then transformed into beats per minute by inverting the average time between beats (in milliseconds) and multiplying by 60,000 milliseconds (the number of milliseconds per minute). The final output is average heart rate in each 10 second window. A graphic example of the average heart rate for a single two minute segment is shown in Figure 3.7. Recall that each point (connected by the line) is an averaged heart rate over a 10 second window.

The second heart variable, heart rate variability, is a little more difficult to calculate. A first order polynomial is fit using ordinary least squares to all time intervals between heart beats in any given 10 second window. Then the slope of this polynomial is used to estimate the change in heart rate. The magnitude of this change can now be used as an estimate of heart rate variability during any 10 second

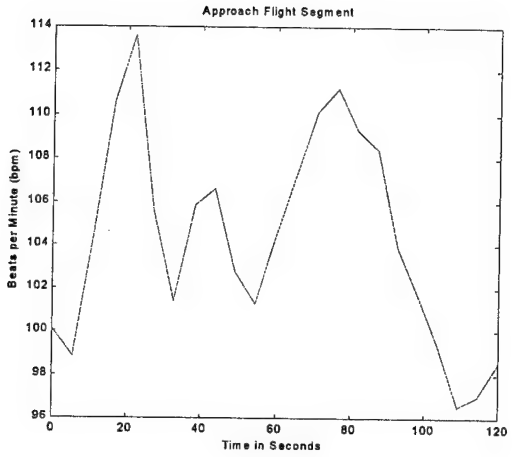


Figure 3.7 Heart Rate

window. The absolute value of the slope of the polynomial serves as the measure of heart rate variability [14]. A graphic example of processed heart rate variability is shown in Figure 3.8.

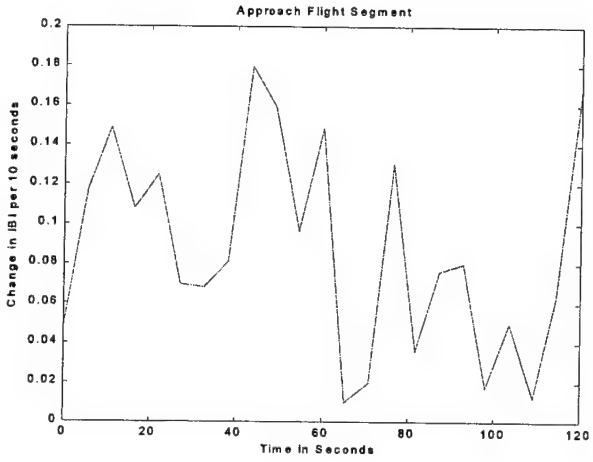


Figure 3.8 Heart Rate Variability

A summary of the steps taken to process the raw heart data provided can be seen in Figure 3.9.

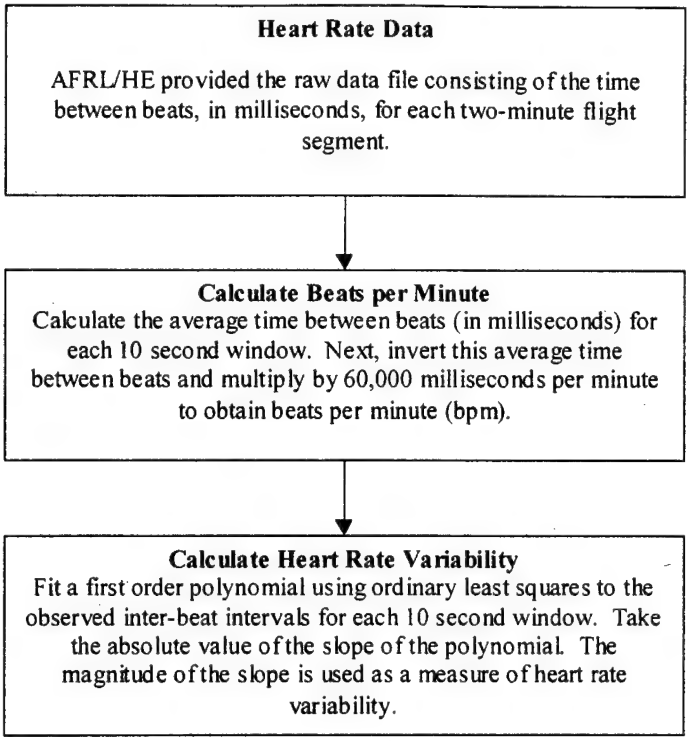


Figure 3.9 Raw Heart Data Processing

*3.4.2 Ocular Measures.* The raw eye data files provided by AFRL/FPL contained three distinct measures of eye movement: blink interval (time in milliseconds between blinks), blink amplitude, and blink duration. Preprocessing yields two physiological ocular features: the number of blinks per time interval, and the average time between blinks. *Matlab* code was written to preprocess the ocular data over the 10 second windows with five seconds of overlap to remain consistent with the EEG data. The number of blinks is calculated by simply identifying and counting the number of blinks in each 10 second window. A graphic example of the number of blinks per 10 second window for one two-minute segment can be seen in Figure 3.10.

The next feature calculated is the average time between blinks. This calculation can be a complicated one. Three scenarios are possible. First, if two or more blinks occur within a 10 second window, the average time between blinks is

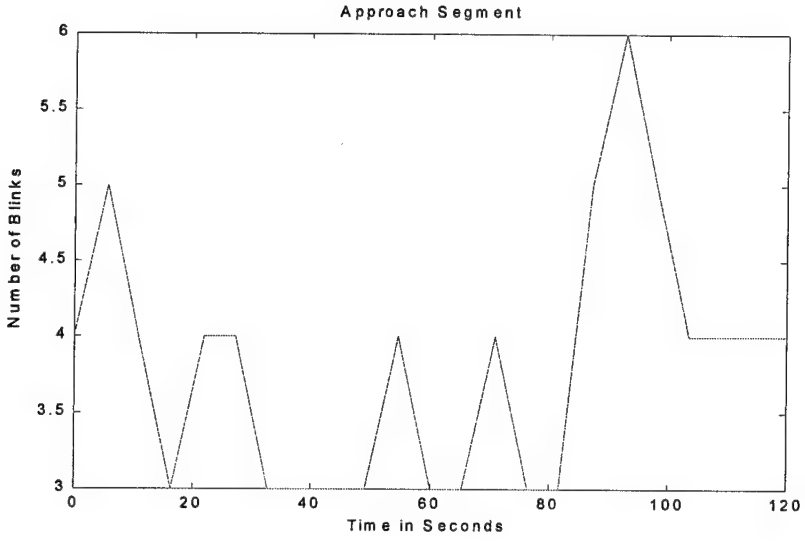


Figure 3.10 Observed Eye Blinks

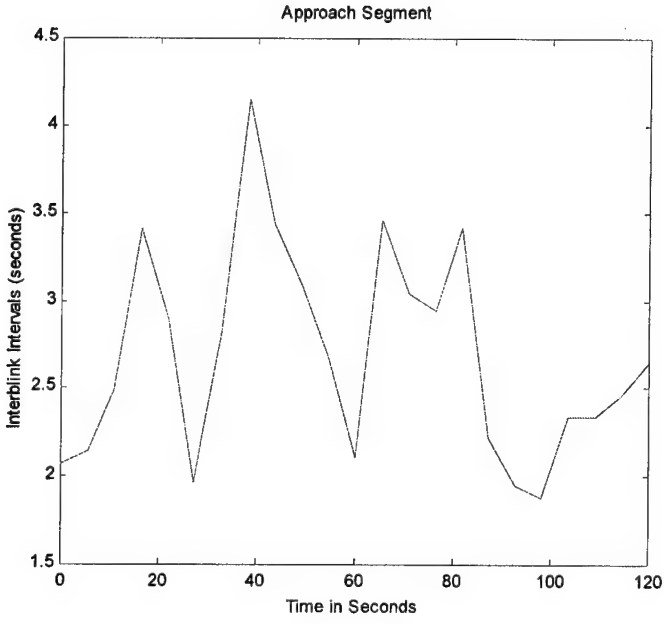


Figure 3.11 Average Time Between Blinks

calculated. Second, if only one blink occurs in a given 10 second window, the prior blink is found and the time between these two blinks is used. Finally, if no blinks occur in a 10 second window the time of the last blink is subtracted from the time at the end of the current 10 second window. In other words, if no blinks occurred, the time recorded is the time the subject has gone without blinking [14]. A graphic example of the inter-blink intervals(IBLIs) is shown in Figure 3.11. Figure 3.12 summarizes the preprocessing done on the raw eye data files.

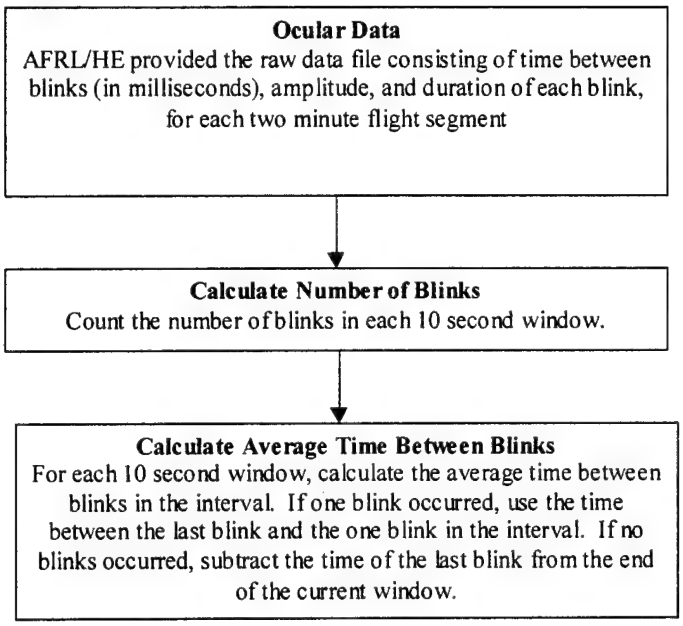


Figure 3.12 Ocular Data Processing

*3.4.3 Respiration Measures.* Two respiration features are processed out of the raw respiration file. The raw respiration file contains the time between breaths (in milliseconds), the minimum breath amplitude and the maximum breath amplitude. The two respiration features obtained are: the number of breaths per unit time, and a measure of the average time between breaths.

These features are processed exactly the same way as the ocular data. The average number of breaths is simply the number of breaths taken per 10 second

interval. The average time between breaths follows the same three scenarios as the eye-blink data. If there are two or more breaths per 10 second window, the average time between the breaths is recorded. If there is only one breath taken in the current window, the time between the current breath and the last breath is recorded. Finally, if there are no breaths are recorded in the current interval, the time of the last breath is subtracted from the end of the current 10 second window. Like the time between blinks, this represents the time the subject has gone without breathing. Figure 3.13 is a representative plot of the number of breaths per 10 second interval over a two minute flight segment. A plot of the average time between breaths over a two minute segment is shown in Figure 3.14. Figure 3.15 is a flow chart containing the procedures for processing the raw respiration data.

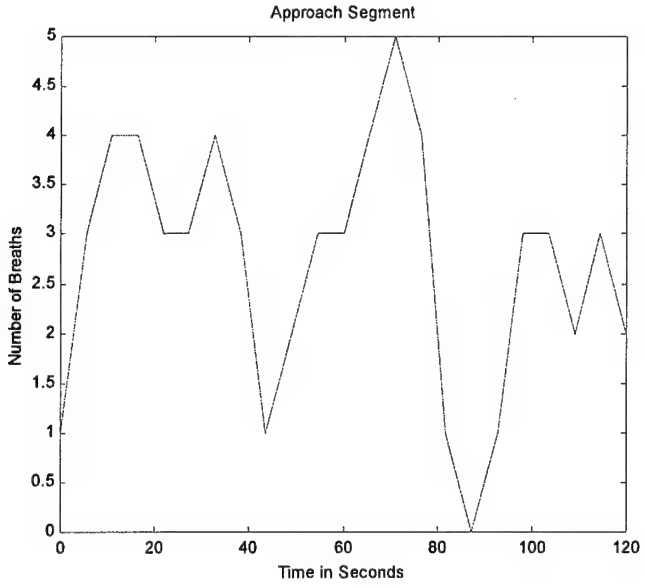


Figure 3.13 Number of Breaths

### *3.5 Summary of Processed Features*

After processing the raw data, a total of 151 psychophysiological features are formed. These features are used to discriminate between mental workload levels.

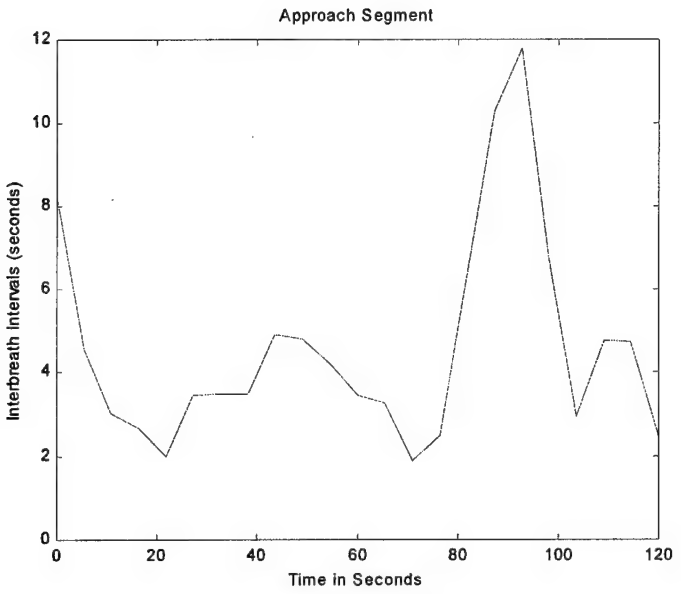


Figure 3.14 Average Time Between Breaths

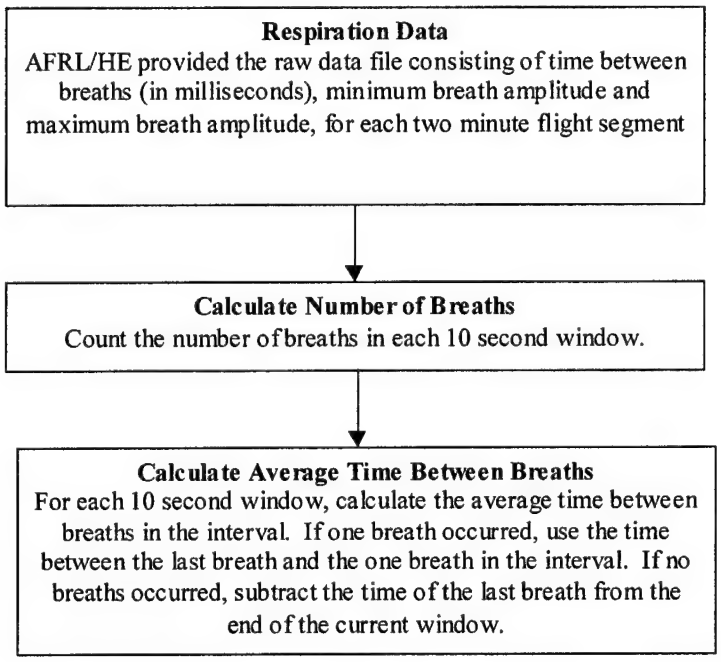


Figure 3.15 Respiration Data Preprocessing

In addition to the 151 features present, a uniform(0,1) random variable is added. This random variable represents random noise used in the signal-to-noise screening method. Recall that a ratio will be used to determine variable contribution by comparing the weights of the input variable to the weights of the noise variable. The smaller the ratio, the less important that input is to the overall classification of mental workload level. Also included as a variable is the actual workload level of the current flight segment. This variable is labeled as 1.0 if the feature belongs to the low/medium mental workload group and labeled as 2.0 if the feature belongs to the high mental workload group. This workload level variable is in the first column simply because the code was written to recognize the first column as the column that contains the identifier for the group the exemplar is associated with. Recall that this is required for artificial neural networks to perform the backpropagation learning method, and is also required for the discriminant analysis to compute the error rate. A truncated version of the final input matrix is shown in Table 3.2.

Table 3.2 Truncated Feature Matrix

| **Feature Number** | **Name** | **Description** | **Units** |
|---|---|---|---|
| 1 | group | 1 if Group 1 / 2 if Group 2 | none |
| 2 | C3d | Power in $\Delta$ Band at C3 | $\log_{10}(\mu V^2)$ |
| 3 | C3t | Power in $\theta$ Band at C3 | $\log_{10}(\mu V^2)$ |
| 4 | C3a | Power in $\alpha$ Band at C3 | $\log_{10}(\mu V^2)$ |
| 5 | C3b | Power in $\beta$ Band at C3 | $\log_{10}(\mu V^2)$ |
| 6 | C3ub | Power in $\mu\beta$ Band at C3 | $\log_{10}(\mu V^2)$ |
| 7 | C4d | Power in $\Delta$ Band at C4 | $\log_{10}(\mu V^2)$ |
| 8 | C4t | Power in $\theta$ Band at C4 | $\log_{10}(\mu V^2)$ |
| 9 | C4a | Power in $\alpha$ Band at C4 | $\log_{10}(\mu V^2)$ |
| 10 | C4b | Power in $\beta$ Band at C4 | $\log_{10}(\mu V^2)$ |
| 11 | C4ub | Power in $\mu\beta$ Band at C4 | $\log_{10}(\mu V^2)$ |
| 146 | hr | Heart Rate | bpm |
| 147 | hrv | Heart Rate Variability | $\Delta$ sec per 10-sec |
| 148 | blnks | Number of Eye-Blinks | # blinks per 10-sec |
| 149 | ibli | Inter-blink Interval | seconds |
| 150 | brths | Number of Breaths | # breaths per 10-sec |
| 151 | ibri | Inter-breath Interval | seconds |
| 152 | noise | Random Uniform(0,1) | none |

### *3.6 Initial Data Inspection*

The statistical program, JMP, was used to investigate the properties inherent in the input data, specifically, correlations between input variables. JMP was also used to find potential outliers in the input data set. A sample of the correlations between a few input variables is shown in Table 3.3.

Table 3.3 Sample Correlation Matrix

| **Correlations** | | | | | |
|---|---|---|---|---|---|
| Variable | C3d | C4d | C5d | C6d | CZd |
| C3d | **1** | 0.4439 | 0.4506 | 0.4324 | 0.4308 |
| C4d | 0.4439 | **1** | **0.9909** | **0.9598** | **0.9691** |
| C5d | 0.4506 | **0.9909** | **1** | **0.9744** | **0.9757** |
| C6d | 0.4324 | **0.9598** | **0.9744** | **1** | **0.9966** |
| CZd | 0.4308 | **0.9691** | **0.9757** | **0.9966** | **1** |

The correlation between two variables is a measure of the linear dependence between the two variables. Positive correlation implies that $X_1$ increases as $X_2$ increases; negative correlation indicates $X_1$ decreases as $X_2$ increases. If the correlation is zero, there is no linear dependence between $X_1$ and $X_2$ [25]. Table 3.3 gives some insight into the correlations between a few of the variables in the input feature set. The bold values indicate the variables that have a high correlation. As the table shows, there is very high correlation between many of the variables. Investigation into the entire EEG feature set reveals that there is high correlation between almost all of the input variables. This information could be useful as an insight into how many features may eventually be kept as classification features.

An investigation into the question of potential outliers was also conducted. The Mahalanobis distance was calculated for each observation. The formula for computing the Mahalanobis distance is as follows:

$$D_i^2 = (x_i - x_{ave})' S^{-1} (x_i - x_{ave}) \tag{3.1}$$

where

$$\begin{aligned} x_i &= \text{the vector of values at observation } i \\ x_{ave} &= \text{the sample mean} \\ S &= \text{the sample covariance matrix} \end{aligned}$$

The Mahalanobis distances are used because they explicitly account for correlations between variables. Figure 3.16 is a plot of the Mahalanobis distances for all observations. Even though there looks to be one outlier in the data set, the data point is close enough to the line that it is not considered a problem.

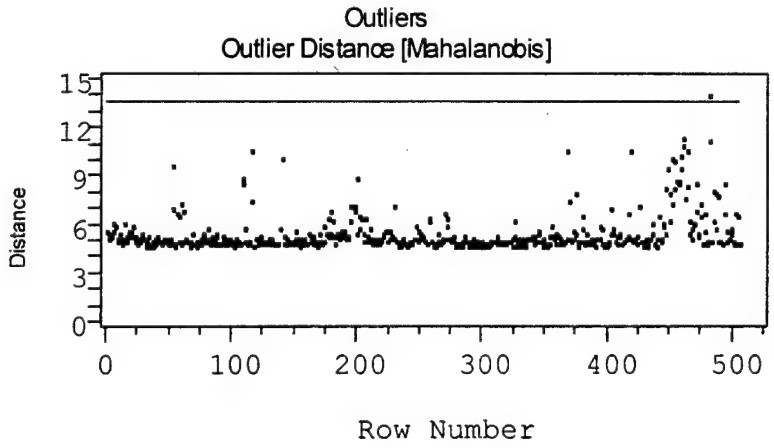


Figure 3.16 Plot of Mahalanobis Distances for all Input Features

### *3.7 Summary of Findings*

This chapter stepped through the preprocessing steps that are needed for, not only the raw EEG data, but also the heart, eye and respiration data. After initial inspection, it looks as though the physiological features indeed react to increased workload level as discussed in the last chapter. Specifically, heart rate increases with increased workload level; eye blinks decrease with increased workload level; the number of breaths tend to increase as mental workload level increases. The next

chapter investigations interbreath, interblink, and interbeat intervals to determine how important these features are in classifying mental workload level. Chapter 4 also presents a methodology to determine which electrodes at which frequency levels are important for classifying mental workload levels. Finally we look at the methodology used in classifying the observations of two pilots over the two days of flight.

# *IV. Methodology and Results for Single Pilot Workload Classification*

This chapter investigates the methodologies used to classify mental workload. These methodologies use the psychophysiological features processed as discussed in the previous chapter. In addition to discussing the initial modeling efforts using discriminant analysis and MLP neural networks, we explore different variable selection efforts. Screening efforts using both discriminant methods and the signal-to-noise ratio (SNR) screening method introduced in Chapter 2 are shown. Finally a factor analysis is conducted on the data set to gain further insight on the variables chosen.

## *4.1 Initial Modeling Efforts*

After preprocessing the raw data, initial efforts are made to determine how well mental workload levels can be predicted. This initial investigation is accomplished using the data for pilot one on day one. This particular data set was used as it was the first available. The following sections examine the methodologies and results of classification on mental workload using a two class discriminant model and a two class MLP neural network.

*4.1.1 Quadratic Discriminant Model.* A description of multivariate discriminant analysis was given in Chapter 2. Quadratic discriminant scores are used to classify a new exemplar as belonging to one of two groups. In our case, group one consists of the low/medium workload segments and group two consists of the high workload segments. Creating the model is a fairly simple process. The first step is to split the data set into a training set and a testing set. This training set is necessary to build the discriminant model. The second step is to gather the necessary components to derive a generic discriminant score for each group. Recall

that the equation for the discriminant scores is,

$$d_i^Q(\underline{X}) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\underline{X} - \widetilde{\mu}_i)^T\Sigma_i^{-1}(\underline{X} - \widetilde{\mu}_i) + \ln P_i \tag{4.1}$$

where

$$\begin{aligned} d_i^Q &= \text{the quadratic discrimiant score for group } i \\ \Sigma_i &= \text{the covariance structure for group } i \\ \underline{X} &= \text{the new exemplar} \\ \widetilde{\mu}_i &= \text{the estimation of mean for group } i \\ P_i &= \text{the prior probability of belonging to group } i \end{aligned}$$

The covariance structures ($\Sigma_i$), the estimation of the mean ($\widetilde{\mu}_i$), and the posterior probabilities ($P_i$), for each group $i$, are determined using the training data set. These are the necessary components for forming the generic discriminant scores for each group. The final step is to determine which group the exemplar belongs to based on the discriminant scores or a comparison of the posterior probabilities.

A discriminant score for each exemplar in the testing data set is obtained for each group using the components obtained from the training set. After all of the discriminant scores have been calculated for each exemplar in the testing data set, the scores are converted to probabilities that will be used for classification of that exemplar into a certain group. This probability is called the posterior probability, and is found using the equation below.

$$P_j(\overline{X}) = \frac{\frac{1}{(2\pi)^{\frac{P}{2}}|\Sigma_j|^{\frac{1}{2}}}\exp[-\frac{1}{2}(\overline{X} - \widetilde{\mu}_j)\Sigma_j^{-1}(\overline{X} - \widetilde{\mu}_j)]}{\sum_{i=1}^{K}\frac{1}{(2\pi)^{\frac{P}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp[-\frac{1}{2}(\overline{X} - \widetilde{\mu}_i)\Sigma_i^{-1}(\overline{X} - \widetilde{\mu}_i)]} \tag{4.2}$$

where

$$
\begin{aligned}
P_j(\overline{X}) &= \text{the posterior probability of class membership} \\
K &= \text{the number of classes} \\
J &= \text{the population number (1 or 2)} \\
\overline{X} &= \text{the new exemplar for classification} \\
\Sigma_j &= \text{the covariance structure for group j} \\
\widetilde{\mu}_j &= \text{the estimation of mean for group j} \\
\pi_j &= \text{the event of belonging to population j}
\end{aligned}
$$

For the two group case $i = j = K = 1, 2$. After the probabilities are calculated for each exemplar, the exemplar is ready to be classified into group 1 or group 2. For each exemplar, the posterior probabilities of belonging to group 1 or group 2 are compared ($p_1(\overline{X})$ compared to $p_2(\overline{X})$). If $p_1(\overline{X}) > p_2(\overline{X})$, the exemplar is classified as belonging to group 1. Similarly, if $p_2(\overline{X}) > p_1(\overline{X})$ the exemplar is classified as belonging to group 2. All exemplars in the testing data set are then classified as group 1 or group 2. A classification accuracy (CA) can be obtained after all exemplars have been classified. The CA is simply the number of exemplars belonging to group 1 classified as group 1 plus the number of exemplars belonging to group 2 classified as group 2 all divided by the total number of exemplars presented from the test data set. The following equation shows the calculation of classification accuracy.

$$CA = \frac{N_{1C} + N_{2C}}{n} \tag{4.3}$$

where

$$\begin{aligned} CA &= \text{classification accuracy} \\ N_{1C} &= \text{the number in group 1 classified as group 1} \\ N_{2C} &= \text{the number in group 2 classified as group 2} \\ n &= \text{the total number of exemplars in test data set} \end{aligned}$$

As mentioned before, the data set from the first pilot on his first day of flight was used to test the quadratic discriminant model. All 151 variables were presented to quadratic discriminant code written in *Matlab*. When an attempt was made to determine how well the classifier performed, we quickly ran into a problem. When attempting to calculate the discriminant scores, we got a warning that the covariance matrices were very close to singular, meaning the determinant was very close to zero (on the order of $10^{-19}$). This minuscule value created problems when classifying the exemplars. The quadratic classifier calculates the log of the determinant of the covariance matrix in the formation of the classifier. If the matrix is close to singular, the determinant is close to zero and we run into the problem of taking the log of zero. The classification accuracy from this run was calculated at 59%. The confusion matrix given in Figure 4.1 shows a real problem. As we can see the

| | | Predicted Group | |
|---|---|---|---|
| | | Group 1 | Group 2 |
| Actual Group | Group 1 | 120 | 0 |
| | Group 2 | 83 | 0 |

Figure 4.1 CA of 151 variables for Quadratic Discriminant Model

classification of exemplars belonging to group 1 is 100% and the classification of the exemplars belonging to group 2 is 0%. This is caused by the near singularity of the

covariance matrix. This problem is thought to be due to the computing accuracy of *Matlab*. It is simply not accurate enough to calculate the determinant of the covariance matrices for a data set with so many highly correlated features. It will be shown later that when the number of features used as inputs is reduced, the quadratic discriminant model works quite well.

*4.1.2 Linear Discriminant Model.* In addition to the quadratic discriminant model, a linear discriminant model can also be used to classify mental workload level. Recall from Chapter 2 that the equation for the linear discriminant score can be calculated as follows:

$$\begin{aligned} d_i^l(\overline{X}) &= \widetilde{\mu}_i^T S^{-1} \overline{X} + w_i \\ w_i &= -\frac{1}{2} \widetilde{\mu}_i^T S^{-1} \widetilde{\mu}_i + \ln(P_i) \end{aligned} \tag{4.4}$$

where

$$\begin{aligned} d_i^l &= \text{the linear discriminant score} \\ \widetilde{\mu}_i &= \text{the estimator of mean for group } i \\ S &= \text{the pooled covariance matrix} \\ P_i &= \text{the prior probability of belonging to group } i \end{aligned}$$

As with the quadratic classification method, the data set is split up into a training and test set. The training set is used to form the estimation of the means for each group ($\widetilde{\mu}_i$), the posterior probabilities ($P_i$) and to form the pooled covariance matrix , Equation 2.23. After calculating these parameters, new exemplars from the test set are presented to the linear discriminant model ($d_i^l$) for classification. As before, a score is computed for each exemplar with respect to the two workload groups. Once all exemplars have linear discriminant scores for both groups, the scores within the groups are converted to probabilities as shown in Equation 4.2.

Once again, the greatest probability for a given exemplar indicates the group the exemplar is classified into.

As mentioned before, the data set from the first pilot on his first day of flight was used to test the linear discriminant model. All 151 variables (including the exemplar's group) were presented to linear discriminant code written in *Matlab*. The data was split 60% for training and 40% for testing the linear discriminant model formed from the training set. Right away we ran into a problem trying to use the linear discriminant function. Like the quadratic function, the covariance matrix was very close to being singular (on the order of $10^{-20}$). This caused enough of a problem that the linear discriminant model was not able to finish running and get a final classification accuracy or confusion matrix. Later it is shown the linear discriminant model does work when the number of input features are reduced.

*4.1.3 MLP Neural Network Models.* All neural network modeling was performed using *Matlab version 5.3* with the *Neural Network Toolbox version 3* according to the techniques outlines in Chapter 2. The MLPs formed for this research are feedforward MLPs with an input layer, a hidden node layer and an output layer. The input layer has one node for every input feature. The output layer contains one node for every output group. The only variable in the neural network is the number of hidden nodes the network will contain. The activation function at the hidden and output nodes is the log-sigmoid activation function. As a reminder, this activation function is given by:

$$f(a) = \frac{1}{1 + e^{-a}}$$

The log-sigmoid activation function will generate outputs from zero to one, as shown in Figure 2.4. All data was also normalized to a mean of zero and standard deviation of one. Finally, the first column of each data matrix indicates the group to which the exemplar belongs. Again we use 1 for low/medium workload level and 2 for high

workload level. Table 4.1 below summarizes the initial MLP architecture. Notice there are two output nodes. The network actually computes probabilities of an exemplar belonging to a certain class. Thus the output of the network will be in the form of percentages between zero and one. In a two class problem, for example,the output will be in a vector form. If the output is classified as group 1, the output vector will look like [1 0]. If the exemplar is classified as group 2, the output vector will look like [0 1]. In actuality, the network will never reach exact determinant values of zero or one. The actual output will look something like [0.9 0.1]. The network then assigns the exemplar to the group with the highest probability, in this example that would be group 1. Technically, only one output node is needed for a two class problem, however this code uses two output nodes.

Table 4.1 Initial MLP Architecture

| **Layer** | **Number of Nodes** |
|---|---|
| Input | 151 |
| Hidden | 151 |
| Output | 2 |

After the initial architecture is set, the training parameters for the network must be determined. For the purposes of just getting the network to run, the weights are initialized to values between -0.05 and 0.05. The neural network uses a batch mode learning method. This means that all exemplars are presented to the neural network. After they all pass through, the error is calculated and all weights and bias terms are updated according to that error, as discussed in Chapter 2. A momentum term is also included to address the problem of getting stuck in local minimums on the error surface. The momentum term was set to 0.9. Other parameters that need to be set for network training include the maximum number of epochs to train (set to 1000), the number of early stopping epochs (set to 50), and the number of hidden nodes (initially set to 151). The number of early stopping epochs is used to tell the network to pause and look at the sum of square error (SSE) for the training and internal validation sets. As long as the SSE for both sets is

Table 4.2 Initial Parameter Settings

| Parameter | Setting |
|---|---|
| Weights | -0.05 to 0.05 |
| Learning Rate | Adaptive |
| Momentum | 0.9 |
| Early Stopping Epochs | 50 |
| Maximum Training Epochs | 1000 |

decreasing, the network will continue training. As soon as the network detects an increase in the internal validation SSE while the training SSE decreases, it will stop training. A summary of the parameter settings is shown in Table 4.2.

After the architecture and parameter settings were selected, we addressed the issue of splitting up the data set. The full data set was initially divided using a 60-40 split. That is 60% for training and 40% for testing. The training data set was further divided up 50-50, meaning, 50% for training and 50% for internal validation. In Chapter 2 it was mentioned that it is common for the test set to be somewhere between 25-30% of the original data set. Since the data sets were not very large, the testing set was kept a little larger than normal. Additionally, since the number of group 1 and group 2 exemplars were not equal, when the data sets were spilt up into training, testing and validation sets, the *Matlab* code made sure to keep the same proportions of group 1 to group 2 equal in all data sets that were formed.

The MLP neural network worked well compared to the quadratic and linear classification efforts. The network stopped training after 113 epochs. Figure 4.2 shows how the internal validation SSE started to level off after about 60 epochs. The network probably trained for a longer time because the early stopping epoch check was ordered every 50 epochs.

Figure 4.2 illustrates the training SSE, internal validation SSE and the test SSE. We expected the test SSE to be a little larger than the internal validation SSE because that data set is totally independent from the data set that was used to form the neural network. The classification accuracy for the test data set was 81.28%.

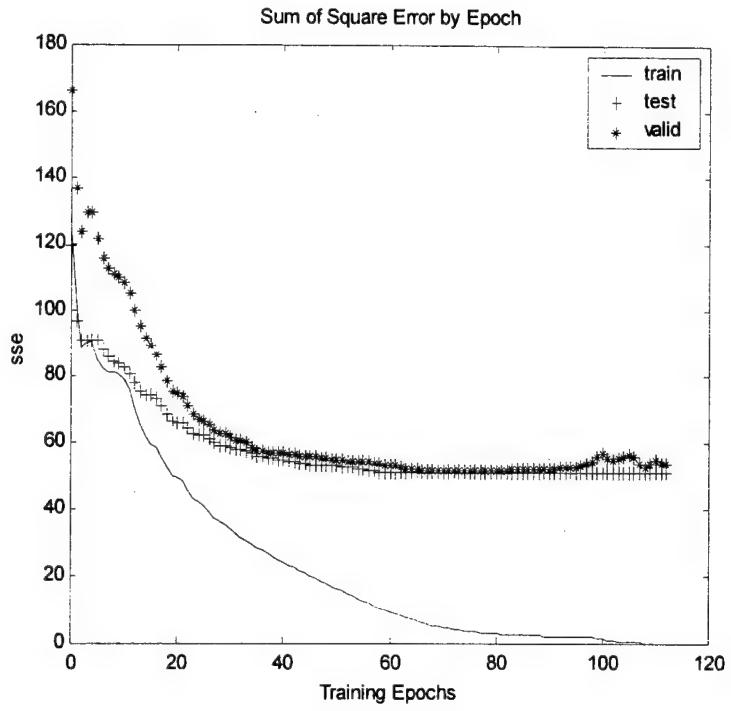


Figure 4.2 Initial MLP Training

This is a marked improvement over the 59% obtained with the quadratic discriminant model. Figure 4.3 illustrates the confusion matrix for the MLP classification. As we can see, the neural network predicted group 1 accurately 85% of the time and predicted group 2 accurately 76% of the time.

*4.1.4 Summary of Initial Efforts.* Three models were used as classifiers for the two group problem. Group 1 is low/medium mental workload level and group 2 is high mental workload level. All models were presented the full 151 features from the data set. At first inspection it seems that the MLP may be the best model for our classification efforts. The MLP gave a decent classification accuracy of 81% while the quadratic model only output a 59% CA. Recall that the linear classifier did not work at all for all 151 features presented. Even though it seems that the

| | | Predicted Group | |
|---|---|---|---|
| | | Group 1 | Group 2 |
| Actual Group | Group 1 | 102 | 18 |
| | Group 2 | 20 | 63 |

Figure 4.3 Confusion Matrix for Initial MLP

MLP may be the best model for classification, the reduction of features will present better insight into how well the quadratic and linear models perform.

## *4.2 Feature Screening Efforts*

As with the initial modeling efforts, both discriminant methods and neural networks are used in screening features out of the original 151 included in the data set.

*4.2.1 Discriminant Screening Effort.* The discriminant feature screening method was accomplished using the *SAS version 6* program. One of the options in *SAS* is to run a procedure called STEPDISC on the input data file. What this procedure does is it takes every input feature and considers each feature for entry into an "optimal" feature set. The procedure passes through the entire data set and, based on a set of p-values and a criteria that those p-values must meet, selects one variable for entry into this "optimal" feature set. It then goes through the entire data set again, minus the variable it picked during the first pass, and selects another variable for entry. The STEPDISC procedure iterates through this process until no p-values meet the specified criteria. This implies that no more variables can be entered into the "optimal" feature set. Procedure STEPDISC can be run one of three ways, forward, backward and mixed. The method that was used in this research effort was the forward method. An important note of interest is that *SAS*

is very sensitive to how the data set is set up as far as the number of spaces between columns, carriage returns and so forth. Because of this sensitivity, the data set for entry into the *SAS* program was passed through a small *Fortran* code that simply took out any extra spaces and carriage returns present in the data. The Fortran code is presented in Appendix C. STEPDISC was first performed on the data set for the pilot 1 on the first day. The results show a dramatic drop from 151 to 34 variables. These variables are listed in Table 4.3.

Table 4.3 Variables Left After SAS Screening Procedure

| **Variable** | **Variable** | **Variable** |
|---|---|---|
| C3a | CZub | P4ub |
| C3u | F3d | P8d |
| C4t | F3b | P9d |
| C4a | F3t | P9ub |
| C4ub | F4d | PO4a |
| C5b | F4a | PO4b |
| C6d | F4b | PZt |
| C6t | F8d | PZa |
| C6a | F8t | PZub |
| C6ub | FC2a | HR |
| CZt | O2d | BLNKS |
| | | BRTHS |

In order to get a good feel of how any particular classifier is doing (linear, quadratic or MLP) we need to be fairly confident about the classification accuracy's each model is reporting. We want to get a 95% confidence interval about the mean using $n$ runs of a particular model. The amount of runs we want to run is driven by the central limit theorem (CLT). The CLT theorem states that the probability distribution for our mean classification accuracy is approximately normal when the sample size is "large". The question is, what exactly is large? It has been shown that the distribution for the mean approaches normality as the sample size approaches $n = 30$ or larger [25]. Therefore, if we run each classifier 30 times, we can get a 95% confidence interval about the mean classification accuracy using the normality assumption to calculate the lower and upper bounds of the confidence interval. The

lower and upper bounds for a 95% confidence interval when $n = 30$ are shown in the formula below.

$$CI = \overline{CA} \pm z_{\alpha/2}(\frac{s}{\sqrt{n}}) \tag{4.5}$$

where

$$\begin{aligned} \overline{CA} &= \text{the mean classification accuracy over } n \text{ runs} \\ z_{\alpha/2} &= \text{the z-value from normal tables} \\ \alpha &= \text{the desired confidence level (0.05 for 95\% CI)} \\ s &= \text{the standard deviation of observed classification accuracies} \\ n &= \text{the number of runs} \end{aligned}$$

All confidence intervals will be calculated in this manner.

*4.2.1.1 Linear and Quadratic Classification.* The purpose of screening is to be able to accurately predict mental workload level, with a minimal loss of classification accuracy. Recall, the classification accuracy was not very good for either the linear or quadratic discriminant models with all 151 variables included. These models were run again, using the 34 variables *SAS* printed out as important. The original data set was modified to include only the 34 important variables and a column identifying to which group the exemplar belongs. As mentioned above, 95% confidence intervals were calculated for both the linear and quadratic discriminant classifiers. The results of these classifiers are shown in Table 4.4.

Table 4.4 Average CA for Pilot 1, Day 1 using SAS variables

| **CI Measure** | **Linear** | **Quadratic** |
|---|---|---|
| Upper 95% limit | 81.337 | 82.20803 |
| **Mean** | **81.4815** | **82.3280** |
| Lower 95% limit | 81.62596 | 82.44806 |

As we can see, the linear and quadratic classifiers both performed much better on the reduced set of variables then on the original set of variables. Statistically, the quadratic classifier performs better, however, the difference is not huge. Either method could be used and be considered a fairly good classifier.

*4.2.1.2 MLP Classification.* A new MLP was created using the 34 variables suggested by the *SAS* STEPDISC procedure. The MLP setup was similar to the initial MLP approach. In addition to trying the MLP on 34 variables to obtain classification accuracy, the number of hidden nodes also varied. This was done to see how classification accuracy is affected by the number of hidden nodes. Since there is no set algorithm for the number of hidden nodes in a neural network, the number of hidden nodes was determined using Kolmogorov's Theorem, and the upperbound method as discussed in Chapter 2. Kolmorogorov's Theorem states that the number of required hidden nodes is never more than twice the inputs. This reasoning led to 34 and 64 hidden nodes, as shown in Table 4.5. The upper bound approach, using Equation 2.1, yields

$$H < \frac{0.5P - 1}{M + 1} = \frac{0.5 * (152) - 1}{34 + 1} = 2.14$$

Recall that $P$ is the number of exemplars in the training set and $M$ is the number of input features. Therefore, the third set of hidden nodes will be equal to 2. Table 4.5 shows 95% confidence intervals on the mean classification accuracy using the MLP with these proposed number of hidden nodes.

Table 4.5 MLP Classification with 34 Input Features and Varying Hidden Nodes

| **Number of Hidden Nodes** | **Lower 95% Limit** | **Mean** | **Upper 95% Limit** |
|---|---|---|---|
| 2 | 80.7865 | **80.9392** | 81.0917 |
| 34 | 82.3446 | **82.5132** | 82.6817 |
| 68 | 82.4424 | **82.5794** | 82.7162 |

Table 4.5 shows that the networks with 34 and 68 nodes perform about the same. Since the confidence intervals for the 34 and 68 hidden nodes overlap, we can say that the mean classification accuracy for these two MLPs are statistically equal. Notice that the MLP with 2 hidden nodes performs around 81%. Although this is lower than the other two, a decision has to be made as to whether or not this difference is important enough to warrant the addition of 32 additional hidden nodes. Looking at the average classification accuracy, another thing that we notice in Table 4.5 is that there really is not too much of a difference between the linear, quadratic or MLP classifiers for this set of data. It looks at though no matter which one we use, we are going to get a fairly good classification accuracy for the data set.

*4.2.2 Signal-to-Noise Screening Effort.* In addition to using a discriminant analysis to pick which variables were important to the problem, a SNR screening method was also used. The data set consists of the 151 input features, a column indicating group membership and an extra column that is a random uniform(0,1) noise feature which will be used in the SNR screening method. The algorithm for the SNR screening method can be found in Chapter 2.

Using the SNR screening method is more of an "art form" than the stepwise discriminant method that *SAS* uses. The stepwise discriminant function uses statistical methods to determine a salient set of input features. This set of input features will not change no matter how many times the set is presented to the STEPDISC procedure. The SNR screening method works a bit differently. Step 12 in the SNR algorithm says to compare the reaction of the test classification error rate to the removal of the individual features. In order to determine how many features to keep we look at a plot of the classification accuracy versus the number of input features. What we look for is a drop off in the classification accuracy. Often times there is not a clear spot where this cut off will be drawn. Figure 4.4 shows an example of this situation.

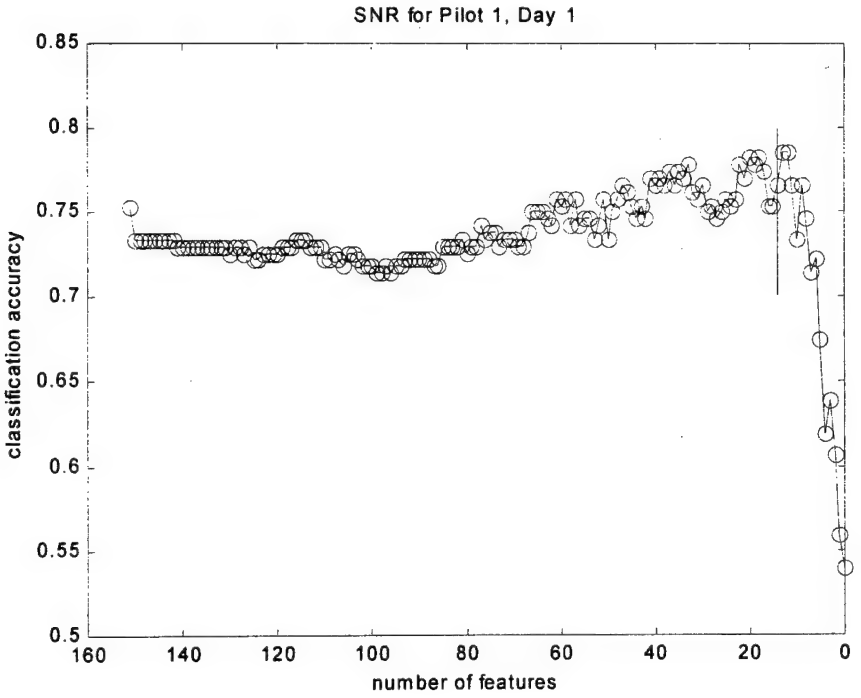


Figure 4.4 SNR Screening for Pilot 1, Day 1

Notice that the classification accuracy dips down and then rises up again at about 28 features and again at 18 features. We must determine where to draw the cutoff for the number of features. In this case the number of features were cut off at 14. A vertical line is drawn on Figure 4.4 to indicate this cutoff point. The number of features picked using the SNR screening method is a decrease from the 34 variables the stepwise discriminant method picked. The 14 variables picked by the SNR are listed in Table 4.6.

It is interesting to note that 10 of the 14 variables selected using the SNR method match the subset using STEPDISC. We will look at this later. Linear, quadratic and MLP classifiers were all used to determine how well the 14 variables picked by the SNR screening method classified pilot mental workload. Once again, each classifier ran thirty times to obtain a confidence interval about the true classi-

Table 4.6 SNR Variables

| Variable | Variable |
|---|---|
| C6d | P9b |
| C6a | PO4a |
| C6b | PO4b |
| C6ub | PZt |
| CZa | T8ub |
| P4ub | HR |
| P8d | BRTHS |

fication accuracy. The results for the linear and the quadratic classifiers are shown in Table 4.7.

Table 4.7 Average CA for Pilot 1, Day 1 using SNR variables

| CI Measure | Linear | Quadratic |
|---|---|---|
| Upper 95% Limit | 73.7605 | 78.2081 |
| **Mean** | **73.9418** | **78.3862** |
| Lower 95% Limit | 74.1231 | 78.5644 |

Notice the drop in classification accuracy using the 14 SNR variables compared to the 34 used by the SAS discriminant method. In addition to the linear and quadratic discriminant classifiers, an MLP was also used with the 14 SNR variables. The structure of the neural network is unchanged from the initial setup except for the number of input features and hidden nodes. Once again the number of hidden nodes varies. The first number of hidden nodes is simply equal to the number of input features. The second number of hidden nodes is obtained using the equation for an upperbound on hidden nodes (Equation 2.1). The results are:

Table 4.8 MLP CA using 14 Input Features with Varying Hidden Nodes

| Number of Hidden Nodes | Lower 95% Limit | Mean | Upper 95% Limit |
|---|---|---|---|
| 8 | 81.2465 | **81.4418** | 81.6371 |
| 14 | 81.5587 | **81.7857** | 82.0127 |

Once again, notice that there is no statistical difference in the results based on the number of hidden nodes used for classification. Furthermore, we see that using

14 variables, the MLP classified with practically the same accuracy as with the 34 variables. This is very interesting since the linear and quadratic classifiers did not fair as well (Table 4.7). This result suggests that when the number of input features was large, the separation of the different groups (high and low) were fairly linear in nature, something both the linear and quadratic classifiers could handle. When the number of features decreased significantly, the data set begins to twist out of that linear state, and the neural network is the only classifier that has a structure that allows adaptation to such a data set.

Recall that the SNR approach found 10 of the 14 variables are contained in the set that *SAS* picked. A question arises from this situation: Why aren't all the SNR variables contained in the set that *SAS* picked? Recall from Chapter 3 that the variables have a large amount of correlation between them. This degree of correlation suggest the variables have some underlying, unknown factor in common. In order to study the inter-relationships of the variables with possible underlying factors, a factor analysis was conducted on the variables picked by the stepwise discriminant method combined with the variables picked by the SNR screening method.

*4.2.3 Factor Analysis.* The idea behind factor analysis is that all variables in a data set are explained by some group of underlying factors. This means that although the data set from the pilot workload study has 151 variables in reality there may be only, say 10, underlying factors. Each variable in the data set has a certain amount of variance that is associated with it. Factor analysis assumes that some of this variance is due to some common variance (how the variable covaries with each factor) and some unique variance that is specific to the individual variable [3]. Figure 4.5 gives a pictorial concept of factor analysis.

Using the FACTOR procedure in *SAS*, a factor analysis was performed on the entire data set. The analysis was done in *SAS* using the FACTOR procedure. This procedure allows manual or automatic selection of the number of factors desired.

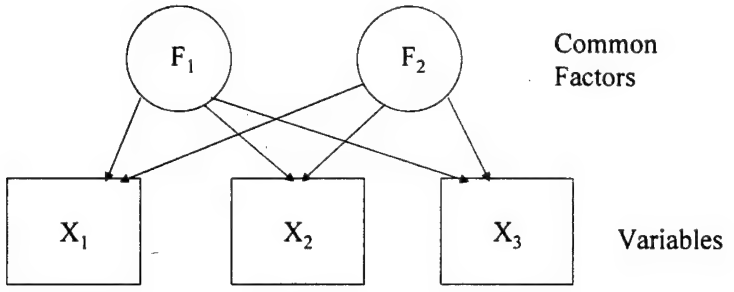


Figure 4.5 Pictoral View of Factor Analysis

This automatic selection is based on a minimum eigenvalue criteria. When factor analysis is first performed on a data matrix, the output contains a certain number of factors. These factors form a space that each variable is contained in. By a space we mean that, if there are two factors, each variable is in two-space and has an associated ($F_1$,$F_2$) value associated with it. This can be thought of as (x,y) coordinates for a point in Euclidean space. If there are three factors output from the factor analysis, each variable is located in three-space and has an associated ($F_1$,$F_2$,$F_3$) value associated with it. These values are the factor loadings of each variable. We can think of each variable as being projected on one of the main axes (i.e., $F_1$ or $F_2$, etc.). This projection indicates how much that variable's variance is explained by that underlying factor. So, if the factor loading for a variable is high, the variable is highly correlated with some underlying, unknown factor. Conversely, if the factor loading is low, the variable is not highly related to that factor. Figure 4.6 presents a pictorial representation of this concept. This figure only represents two-space as an example but can be easily expanded to encompass higher dimensions.

When factor analysis is first performed, the loadings can be somewhat ambiguous. It sometimes is not clear which variables are associated with which factors. Consider the following example. Factor analysis is performed on three variables with the results shown in Table 4.9. Notice that on first inspection it looks like the variable $X_1$ is associated with factor 1 and $X_2$ is associated with factor 2. However, when we try to determine which factor $X_3$ is explained by, it is difficult to say because the

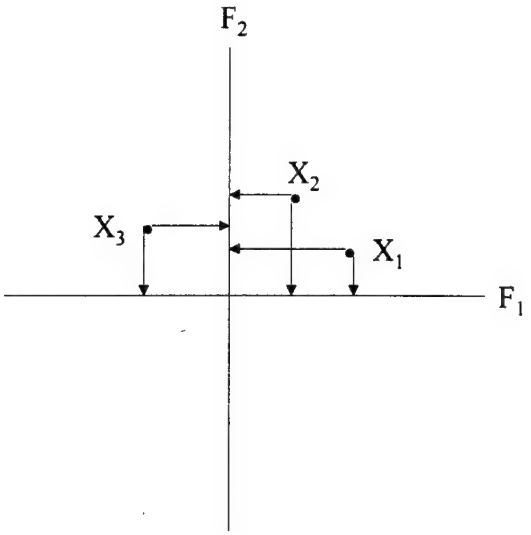


Figure 4.6 Initial Factor Analysis

Table 4.9 Hypothetical Initial Factor Analysis

| **Variable** | $\mathbf{F_1}$ | $\mathbf{F_2}$ |
|---|---|---|
| $X_1$ | **0.8** | 0.2 |
| $X_2$ | 0.3 | **0.7** |
| $X_3$ | 0.5 | 0.5 |

factor loadings spread it equally across both factors 1 and 2. This result is difficult to interpret. In order to be able to interpret the actual loadings, an orthogonal rotation of the space formed by the factor axis can be performed. Theoretically, variance of a data set and the factor solution of that data set does not change after one rigid rotation. A rigid, or orthogonal, rotation maintains a $90^0$ angle between all axes. Figure 4.7 shows an orthogonal rotation of the factor axes. Once again this example is shown in two-space. The most common orthogonal rotation scheme is called the varimax rotation and is an option in the FACTOR procedure in *SAS*. The factor loadings after the varimax rotation are shown in Table 4.10. Now we can clearly see that $X_1$ and $X_2$ are both related to some underlying common factor, $F_1$ and $X_3$ is related to an independent factor, $F_2$. Now that we have an understanding of factor analysis, we can apply the technique on the pilot mental workload data set.

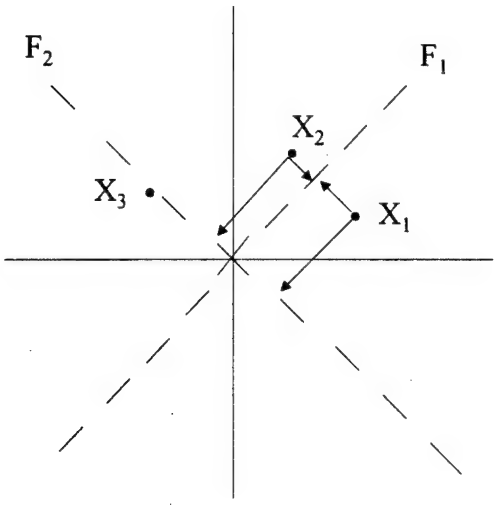


Figure 4.7 Orthogonal Rotation of the Factor Axes

Table 4.10 Factor Loadings After Orthogonal Rotation

| Variable | $F_1$ | $F_2$ |
|---|---|---|
| $X_1$ | **0.9** | 0.1 |
| $X_2$ | **0.85** | 0.15 |
| $X_3$ | 0.25 | **0.75** |

*4.2.3.1 Factor Analysis on Screened Variables.* We use a factor analysis on the entire data set to determine how the two variable sets (one from the stepwise discriminant analysis and one from the SNR screening method) relate to each other. The proposition is that even though the 34 variables from the stepwise discriminant method did not contain all of the 14 variables from the SNR screening method, the factor analysis will show that there are some unknown common factors linking the variables from both methods together. The entire data set, pilot 1 on day 1, was presented to the *SAS* FACTOR procedure using the varimax orthogonal rotation option. The results are presented in Table 4.11.

The factor analysis on the screened variables implies that there might be seven underlying factors driving the selection of the important factors for workload classification. This result helps explain why the two screening methods (discriminant and SNR) chose different variables in some cases. Even though different variables

Table 4.11 Factor Analysis for Pilot 1, Day 1

| | $F_1$ | $F_2$ | $F_4$ | $F_7$ | $F_8$ | $F_{10}$ | $F_{11}$ |
|---|---|---|---|---|---|---|---|
| **C3a** | *0.91925* | | | | | | |
| **C3ub** | *0.92112* | | | | | | |
| **C4t** | *0.92221* | | | | | | |
| **C4a** | *0.90737* | | | | | | |
| **C4ub** | *0.91235* | | | | | | |
| **C5b** | *0.89917* | | | | | | |
| **C6d** | **0.92472** | | | | | | |
| **C6t** | *0.95489* | | | | | | |
| **C6a** | **0.93599** | | | | | | |
| **C6b** | 0.93869* | | | | | | |
| **C6ub** | **0.94298** | | | | | | |
| **CZt** | *0.94761* | | | | | | |
| **CZa** | 0.94571* | | | | | | |
| **CZub** | *0.91677* | | | | | | |
| **F3d** | *0.9408* | | | | | | |
| **F3t** | *0.94062* | | | | | | |
| **F3b** | *0.92287* | | | | | | |
| **F4d** | *0.91212* | | | | | | |
| **F4a** | *0.9074* | | | | | | |
| **F4b** | *0.90508* | | | | | | |
| **F8d** | *0.81474* | | | | | | |
| **F8t** | *0.81648* | | | | | | |
| **FC2a** | *0.93462* | | | | | | |
| **O2d** | *0.93349* | | | | | | |
| **P4ub** | **0.54652** | **0.68469** | | | | | |
| **P8d** | | **0.65732** | **0.56736** | | | | |
| **P9d** | *0.62883* | *0.62791* | | | | | |
| **P9b** | | 0.73838* | | | | | |
| **P9ub** | | *0.90234* | | | | | |
| **PO4a** | | | | | | **0.69661** | |
| **PO4b** | | | | | | **0.72042** | |
| **PZt** | | **0.7172** | **0.61551** | | | | |
| **PZa** | | *0.91157* | | | | | |
| **PZub** | | *0.91504* | | | | | |
| **T8ub** | | 0.86147* | | | | | |
| **HR** | | | | **0.77191** | | | |
| **BLNKS** | | | | | *0.90791* | | |
| **BRTHS** | | | | | | | **-0.79302** |

were reported as important, notice that all of these variables lie in similar dimensions. This shows that even though the screening efforts give differing results, the same basic dimensions that drive the data set are being covered.

In order to help interpret the factor analysis, we consulted Dr. Glen Wilson, head of the research effort at AFRL/HE. First we had to understand the functions of the various parts of the brain. Figure 4.8 is a representation of the electrode placement used in this experiment. The nomenclature for the electrodes, as intro-

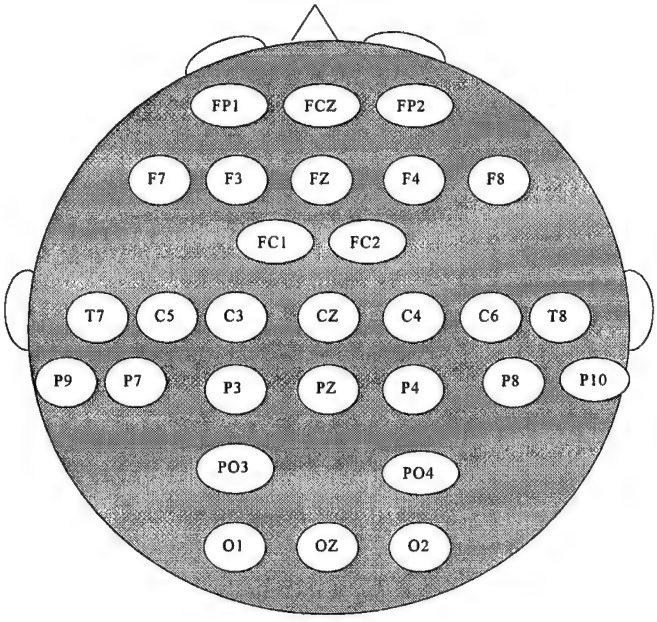


Figure 4.8 Electrode Placement

duced in Chapter 3, is: F - frontal, C - central, T - temporal, P - parietal, and O - occipital. The electrodes are placed in such a manner as to try and capture what the brain goes through when mental workload level increases. The frontal area of the brain (covered by the electrodes that start with F) is where planning activities and higher order cognitive functions occur. It is the decision making area. The central (C) part of the brain drives motor functions, such as moving legs, feet, hands, etc. The temporal (T) and occipital (O) portions of the brain are associated with auditory and visual functions, respectively. Finally, the parietal (P) portion of the

brain is the association area. An example of association would be looking at an apple. We look at the apple and the vision is processed in the occipital area. This processed signal leaves the occipital area and travels through to the parietal area where our brain then tells us that the thing we are looking at is an apple.

Now that we have an understanding of how each area of the brain works we can try to draw conclusions as to what each factor means. Figure 4.9 shows a scheme of the electrodes contained in Factor 1. The light shaded electrodes indicate variables that were picked by both the *SAS* discriminant screening procedure and the SNR screening procedure. The darker electrodes are variables that were picked only by the *SAS* procedure. Notice that most of the variables are in the central and frontal area. The frontal area is associated with planning and the central with motor skills. It is possible that the first factor can be explained by planning actions during flight and the muscular movements associated with these actions that need to be performed. A good summary might be that factor 1 is associated with decision making and the actions performed as a result of those decisions.

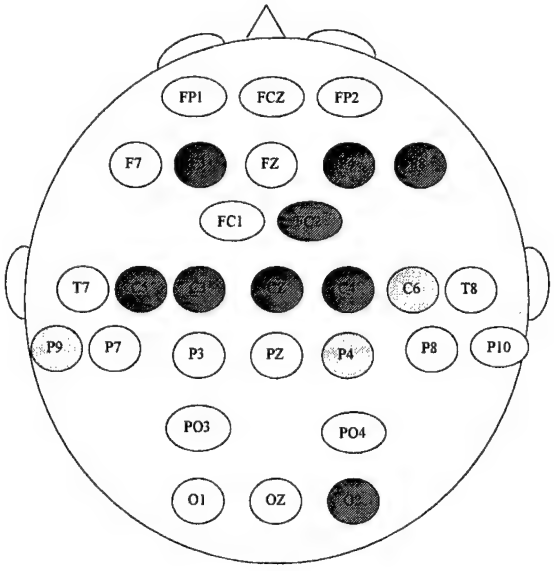


Figure 4.9 Factor 1 for Pilot 1, Day 1

Notice in Table 4.11 that factor 4 is a complete subset of factor 2. Figure 4.10 illustrates all electrodes in factors 2 and 4. Once again, the darker shades indicate the electrodes that are picked by only the *SAS* stepdiscrim procedure. The lighter shaded electrodes indicate the variables that were picked by both the *SAS* discriminant procedure and the SNR screening procedure. Notice that these electrodes are generally located in the parietal portion of the brain. Recall that the parietal region is the region associated with associations. Therefore, we can conclude that factors 2 and 4 might be driven by some lower level association process.

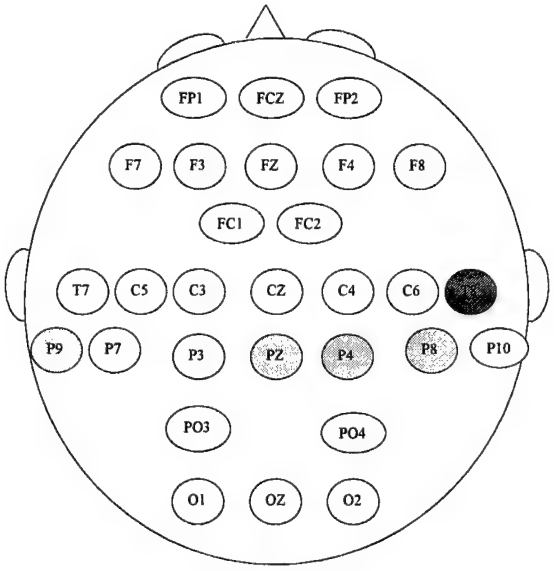


Figure 4.10 Factors 2 and 4 for Pilot 1, Day 1

Referring back to Table 4.11, notice that factor 10 contains the electrode PO4. This may indicate that this factor somehow related to vision and association. The other three factors each contain only one peripheral measure each. It is clear to see that factor 7 is associated with heart measures, factor 8 is associated with eye measures, and factor 11 is associated with respiratory measures. Therefore we have possible explanations for each underlying factor.

## *4.3 Summary of Findings*

This chapter has presented classification efforts for pilot mental workload. Initial efforts used both discriminant and neural networks, and all 151 variables. Initial screenings were conducted on the variables using the *SAS* discriminant procedure, STEPDISC, and the SNR screening feature using an MLP. Our first observation was that the discriminant method chose to retain more variables than the SNR screening method. Secondly we find that almost all the SNR variables were contained in the set chosen by the discriminant procedure. In order to gain some insight on why the different methods chose different variables, factor analysis was conducted on the data set. The factor analysis revealed that all the variables, from both *SAS* and the SNR screening method, could be explained by 6 or 7 underlying factors. This means that even though the methods were choosing different variables, the same underlying, driving factors were found. Next we attempted to interpret what these underlying factors represent. Using the knowledge of the workings of the brain, each factor was coupled with a possible explanation.

An additional insight was gained from the factor analysis. As mentioned before, the factor analysis showed how each variable, whether from the *SAS* screening method or the SNR screening method, lay on one of 7 factors. Additionally, every factor contained at least one variable that was in both the *SAS* and SNR variable sets. Once again linear, quadratic and neural classifications models were used to get an estimate of the classification accuracies using the variables contained in both the *SAS* and the SNR screening methods. The variables used in this classification effort are listed in Table 4.12.

The following table gives an overall summary of this classification effort as well as the previous classification efforts. This gives a clear representation and summary of how classification accuracy is affected by the number of input variables. The MLP structures all contain the number of hidden nodes equal to the number

Table 4.12 Common Variables from SAS and SNR, Pilot 1, Day 1

| Variable | Variable |
|---|---|
| C6d | PO4a |
| C6a | PO4b |
| C6ub | PZt |
| P4ub | HR |
| P8d | BLNKS |
| P9d | |

of input features. Recall that classification with all 151 variables was not possible using the linear classifier.

Table 4.13 Summary of Analysis for Pilot 1, Day 1

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 151 vars | Lower | N/A | N/A | 78.33 |
| | Mean | | | **78.50** |
| | Upper | | | 78.67 |
| SAS 34 vars | Lower | 81.34 | 82.21 | 82.34 |
| | Mean | **81.48** | **82.33** | **82.51** |
| | Upper | 81.63 | 82.45 | 82.68 |
| SNR 14 vars | Lower | 73.76 | 78.21 | 81.56 |
| | Mean | **73.94** | **78.39** | **81.79** |
| | Upper | 74.12 | 78.56 | 82.01 |
| Factor Analysis 10 vars | Lower | 74.10 | 74.57 | 77.55 |
| | Mean | **74.25** | **74.78** | **77.79** |
| | Upper | 74.40 | 74.98 | 78.03 |
| Factor Analysis 6 vars | Lower | 69.7956 | 71.1286 | 73.9074 |
| | Mean | **69.9339** | **71.2566** | **74.0741** |
| | Upper | 70.0722 | 71.3847 | 74.2408 |

Notice in Table 4.15 that there is an additional classification attempt using 6 variables from the factor analysis. This classification effort represents an attempt to use only one variable associated with each factor. There are only 6 variables since the only factors used were those that contained variables chosen by both *SAS* and the SNR screening method. The thought behind this classification method was that each factor identifies a specific dimension that drives the data. It was hypothesized that perhaps we only need one variable from each factor for classification. The

variables were chosen based on the factor loading associated with that factor. The variable with the highest factor loading was deemed the "most important." Notice that for factor 4 in Table 4.11 the highest loading is on variable PZt with a value of 0.61551. However, PZt was also the variable with the highest loading for factor 2. Therefore, the factor chosen to represent factor 4 was P8d. Additionally, since factor 8 did not have a variable shared by both the *SAS* and SNR screening methods, that factor was not represented in the final analysis. Table 4.14 lists the variables used and the factor each represents in the classification analysis using 6 variables.

Table 4.14 Variables Used in Final Factor Analysis

| **Variable** | **Variable** |
|---|---|
| C6ub (F1) | HR (F7) |
| PZt (F2) | PO4b (F10) |
| P8d (F4) | BRTHS (F11) |

The detailed results listed in Table 4.13 were found using data from pilot 1 on day 1. Similar analysis was performed on pilot 1, day 2.

### *4.4 Summary of Analysis for Single Pilots*

The processes for analysis outlined in the previous sections were applied to pilot 1, day 2 and pilot 4 days 1 and 2. Appendix D gives the lists of variables that were used for classification from the *SAS* screening method, the SNR screening method and the factor analysis. The following tables present individual summaries of both pilots, both days. Tables 4.15 and 4.16 are the classification summaries for pilot 1 and Table 4.17 and 4.18 are the classification summaries for pilot 4.

Table 4.15 Summary of Analysis for Pilot 1, Day 1

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 151 vars | Lower | N/A | N/A | 78.33 |
| | **Mean** | | | **78.50** |
| | Upper | | | 78.67 |
| SAS 34 vars | Lower | 81.34 | 82.21 | 82.34 |
| | **Mean** | **81.48** | **82.33** | **82.51** |
| | Upper | 81.63 | 82.45 | 82.68 |
| SNR 14 vars | Lower | 73.76 | 78.21 | 81.56 |
| | **Mean** | **73.94** | **78.39** | **81.79** |
| | Upper | 74.12 | 78.56 | 82.01 |
| Factor Analysis 10 vars | Lower | 74.10 | 74.57 | 77.55 |
| | **Mean** | **74.25** | **74.78** | **77.79** |
| | Upper | 74.40 | 74.98 | 78.03 |
| Factor Analysis 6 vars | Lower | 69.80 | 71.13 | 73.91 |
| | **Mean** | **69.93** | **71.26** | **74.07** |
| | Upper | 70.07 | 71.38 | 74.24 |

Table 4.16 Summary of Analysis for Pilot 1, Day 2

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 151 vars | Lower | N/A | N/A | 75.37 |
| | **Mean** | | | **75.52** |
| | Upper | | | 75.67 |
| SAS 71 vars | Lower | 74.54 | N/A | 78.27 |
| | **Mean** | **74.83** | | **78.51** |
| | Upper | 75.11 | | 78.74 |
| SNR 17 vars | Lower | 75.69 | 78.62 | 76.96 |
| | **Mean** | **75.81** | **78.81** | **77.16** |
| | Upper | 75.94 | 79.01 | 77.36 |
| Factor Analysis 13 vars | Lower | 73.88 | 76.14 | 74.92 |
| | **Mean** | **74.07** | **76.29** | **75.15** |
| | Upper | 74.25 | 76.45 | 75.37 |

Table 4.17 Summary of Analysis for Pilot 4, Day 1

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 146 vars | Lower | N/A | N/A | 97.07 |
| | **Mean** | | | **96.72** |
| | Upper | | | 96.38 |
| SAS 79 vars | Lower | 85.47 | N/A | 97.36 |
| | **Mean** | **85.78** | | **97.46** |
| | Upper | 86.09 | | 97.55 |
| SNR 5 vars | Lower | 86.62 | 90.75 | 91.67 |
| | **Mean** | **86.76** | **90.87** | **91.78** |
| | Upper | 86.91 | 90.99 | 91.90 |
| Factor Analysis 3 vars | Lower | 86.23 | 90.35 | 90.31 |
| | **Mean** | **86.37** | **90.45** | **90.41** |
| | Upper | 86.51 | 90.55 | 90.51 |

Table 4.18 Summary of Analysis for Pilot 4, Day 2

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 151 vars | Lower | N/A | N/A | 86.73 |
| | **Mean** | | | **86.91** |
| | Upper | | | 87.08 |
| SAS 62 vars | Lower | 77.05 | N/A | 90.34 |
| | **Mean** | **77.28** | | **90.50** |
| | Upper | 77.52 | | 90.65 |
| SNR 5 vars | Lower | 75.77 | 82.16 | 85.62 |
| | **Mean** | **76.05** | **82.33** | **85.76** |
| | Upper | 76.33 | 82.50 | 85.90 |
| Factor Analysis 3 vars | Lower | 77.51 | 81.28 | 85.78 |
| | **Mean** | **77.74** | **81.46** | **85.92** |
| | Upper | 77.94 | 81.64 | 86.05 |

There is one interesting point to note from the factor analysis on all pilots. Looking at the individual factor loadings for both pilots, both days, shown in Appendix E, we notice similar results for pilot 1, days 1 and 2. Recall that the italicized variables are the variables that were picked by the *SAS* screening method, the variables with an asterick were picked by the SNR screening method and the bold variables are variables that were picked by both screening methods. Summarizing the results we find factor 1 contains variables concentrated in the frontal and central regions of the brain. Factor 2 contains variables from the parietal and temporal areas of the brain. The peripheral measures (heart rate, eye blinks, etc.) selected their own individual factors. After the factor analysis was done for pilot 1, we expected to see the same results for pilot 4. As we can see in Appendix E, the results were drastically different. Pilot 4 on the first day looks to have three main factors that drive the classification (factors 1, 2 and 3). Another interesting note is the peripheral measures. Eye related measures, (blinks and interblink interval) are

located on their own factor as is interbreath interval. However, heart rate is not on its own factor, it is lumped in with factor 1. This observation is contrary to what we observed with pilot 1, that all peripherals lie on their own factor. An even more perplexing picture is presented when we look at the factor analysis for pilot 4 on day 2. The peripherals are once again on their own factors, however the other factors are puzzling. Factor 1 seems to be important again, however, like pilot 4, day 1, factor 1 seems to contain most of the variables from the parietal region as well as most electrodes from the frontal region. Factor 2 contains variables from the frontal and occipital areas. Finally, factors 3, 4, and 5 contain variables from the central region.

There are a couple of possible explanations for the discrepancies found in the factor analysis. The most feasible reason could be pilot experience. The individual variables could be loading on different factors because of how the pilots react to certain situations. For example, we noticed in the factor analysis on pilot 4, day 1, that heart rate was not on its own individual factor like observed in the factor analysis on all other data sets. This could be due to the fact that stressful situations don't affect this pilot as much as another pilot. While heart rate was still chosen as a significant factor in predicting mental workload level, it is not so significant that it explains a different factor driving the data.

There seems to be no correlation between the factor analysis from pilot 1 and pilot 4. We decided to run a factor analysis on a combined data set. This data set consisted of all data from pilot 1 and all data from pilot 4. One important issue came up when combining the data sets. The data set for pilot 4, day 1 only contained 146 variables. Five variables had to be removed because of bad data that could not be fixed. In order to perform the factor analysis on the entire data set, these same 5 variables had to be removed for the other three data sets. The following table shows the results of this factor analysis.

Table 4.19 Factor Analysis on Both Pilots, Both Days

| | Factor 1 | Factor 2 | Factor 3 | Factor 9 | Factor 11 |
|---|---|---|---|---|---|
| C5ub | B | | | | |
| C6d | A | | | | |
| C6a | A | | | | |
| C6ub | A | | | | |
| CZub | B | | | | |
| F3d | B | | | | |
| FC2t | B | | | | |
| FP2d | B | | | | |
| FP2a | B | | | | |
| FP2b | X | | | | |
| O2d | Y | | | | |
| OZd | B | | | | |
| P3d | | Y | | | |
| P4ub | | A | | | |
| P8d | | A | | | |
| P8ub | | B | | | |
| PO4a | | A | | | |
| PO4b | | A | | | |
| PZt | | A,B | | | |
| T7a | | X | | | |
| HR | | | A,B,X,Y | | |
| BLNKS | | | | | A,B |
| BRTHS | | | | A,B | |

All of the variables on the table are variables that were chosen by both the *SAS* screening method and the SNR screening method, regardless of which data set those variables came from (pilot 1, day 1; pilot 4, day1; etc.). Instead of the factor loadings, the analysis contains which data set that variable came from. A indicates pilot 1, day 1; B indicates pilot 1, day 2; X indicates pilot 4, day 1; and Y indicates pilot 4, day 2.

A nice pattern results from the total factor analysis on all four data sets. As we can see, there seem to be two dominant factors containing the electrodes and all peripheral measures are completely contained on their own factors. Once again we can attempt to give meaning to the factors. The factors containing the peripheral

measures are explained for themselves. Figure 4.11 gives a visual representation of the variables contained on factor 1. Notice that the variables from factor 1 located

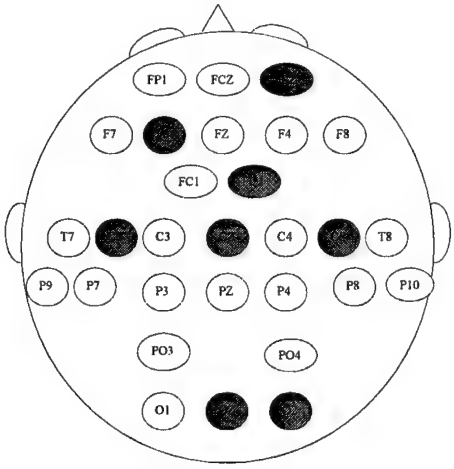


Figure 4.11 Factor 1, Both Pilots, Both Days

in the frontal, central and occipital areas of the brain. This could indicate that the dimension driving factor 1 is associated with higher planning, the motor skills associated with that planning and whatever visual information the pilot is receiving. Looking at the electrodes associated with factor 2 we can come to a similar analysis. Figure 4.12 is a visual representation of the variables contained on factor 2.

The variables that are contained in factor 2 are solely contained in the parietal region and the temporal region of the brain. This could indicate that the second factor is driven by low level association and some auditory measures. We can see that even though the individual analysis for each pilot, each day showed differing results. The overall factor analysis indicates there may be a pattern to the variables that are chosen for classification. Variables from factor 1 and 2 are always chosen and whichever peripheral measure is chosen, they are going to fall on their own factors.

This chapter has dealt with the screening methods and classification efforts on individual pilots, separate days. The next chapter delves into the screening results

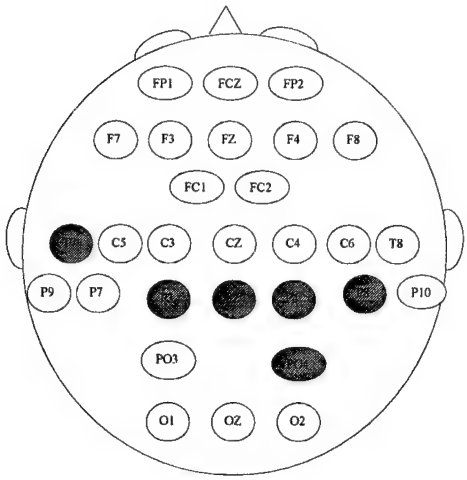


Figure 4.12 Factor 2, Both Pilots, Both Days

and classification efforts for two scenarios. The first scenario is classification across days for a single pilot. The second scenario is classification across multiple pilots, multiple days.

# V. *Classification and Screening Efforts for Multiple Pilots, Multiple Days*

While the last chapter focused on the classification and screening efforts for individual pilots on a single day, this chapter focuses on classification and screening efforts for multiple pilots, multiple days. Section one focuses on classifying mental workload across days for a single pilot. The second section discusses the results using one pilot, both days, to classify mental workload level on another pilot, both days.

## *5.1 Classification for One Pilot, Across Days*

In Chapter 4, classification efforts focused on evaluating individual pilots on one day. The classification results were highly dependent on the pilot analyzed as well as on the day the pilot was analyzed. For example, pilot 4 on day 1 consistently classified in the 90% region with the MLPs. The classification results for pilot 4 on day 2 were slightly lower, in the mid 80% region with the MLPs. This suggests that classification is highly dependent on the particular pilot on that particular day. One hypothesis proposed was to investigate classification efforts for a single pilot, across days. The following results are strictly for pilot 1.

*5.1.1 Screening and Classification Results.* As mentioned above, this classification effort was performed solely on pilot 1. The idea was to use pilot 1, day 1 to form classifiers to predict pilot 1, day 2. Recall in Chapter 4, screening results were presented on pilot 1, day 1 data for the *SAS* discriminant and SNR screening methods, as well as the factor analysis comparison. The variables from these screening results were used for forming the classifiers and attempting to classify mental workload in pilot 1, day 2. Table 5.1 summarizes the classification efforts across days for pilot 1.

Table 5.1 Classification Results for Pilot 1, Across Days

| Analysis | 95% CI | Linear | Quadratic | MLP |
|---|---|---|---|---|
| Initial 151 vars | Lower | **46.31** | N/A | 59.84 |
| | Mean | | | **60.03** |
| | Upper | | | 60.22 |
| SAS 34 vars | Lower | **59.28** | N/A | 59.06 |
| | Mean | | | **59.27** |
| | Upper | | | 59.48 |
| SNR 14 vars | Lower | **50.10** | N/A | 54.65 |
| | Mean | | | **54.84** |
| | Upper | | | 55.03 |
| Factor Analysis 10 vars | Lower | **60.68** | **52.89** | 55.68 |
| | Mean | | | **55.89** |
| | Upper | | | 56.11 |

There is one interesting item to note about Table 5.1. Notice for the linear and quadratic classifiers only one value is reported. In Chapter 4 classification efforts were performed on one pilot, on a single day. In order to take into account any variation in the classification accuracy, the data set was split between training and testing data sets. The training set is used to form the discriminant classifier and the testing set tests how well that discriminant classifier performs. The data is then randomly shuffled and split again between training and testing data sets. This is done 30 times in order to get a confidence interval about the mean classification accuracy. For this portion of the research, the data sets were a bit different. The training set contained all the data from pilot 1, day 1 and the testing set contained all the data from pilot 1, day 2. Even if we shuffle these data sets, the same data is available to form the discriminant classifier and to test the classifier. The discriminant methods don't care if the data is presented in a different order. Therefore, no matter how the data is presented to form the classifier, or to calculate the classification accuracy, the discriminant classifiers will report the same classification accuracy every single time. This trend does not hold in the case of the neural network.

The network can be thought of as a semi-living being. Even though the data presented for forming the network and for calculating classification accuracies doesn't change, there is enough subjectivity associated with the neural network that the answers will be slightly different each time the data sets are presented. For example, the entire data set for pilot 1, day 1 is presented to form a neural network. The network is formed resulting in a certain number of input nodes, hidden nodes, output nodes and weighted connections between each layer. The initial weighted connections are random numbers that change each time the data set is presented to form the neural network. Herein lies the subjectivity of the neural network. This randomness results in slight changes in the final weighted connections for each neural network that is formed. The slight changes in the weights result in varying classification accuracies on the test set, pilot 1, day 2.

## *5.2 Classification Across Pilots*

Thus far we have investigated classification efforts of individual pilots on one day (Chapter 4) and for an individual pilot across days. The next step is to investigate the hypothesis of forming one classifier that will perform adequately regardless of the pilot and regardless of the day. This hypothesis was tested using all data from pilot 1 to form the classifiers and all data from pilot 4 as new exemplars for classification. One modification had to be made to the data set before classification could begin. Recall from Chapter 4 that the data set from pilot 4, day 1 contained only 146 variables. Five EEG variables had to be removed because of bad sections. Because of the reduced data set from pilot 4, day 1, the same three variables were removed from pilot 1, days 1 and 2 and pilot 4, day 2 data sets.

*5.2.1 Screening and Classification Results.* An attempt was made to reduce the total number of variables required for classification. As presented in the previous chapter, both the *SAS* STEPDISC procedure and the SNR screening methods were used to reduce the number of variables. Additionally, factor analysis was

also used to draw a comparison between the variables using the discriminant *SAS* screening method and the SNR screening method. The variables that were picked by both the *SAS* and SNR screening methods were also used for classification. After the variables were acquired from the different screening methods, the data set was presented to each classifier, as was done in the previous chapter. Table 5.2 summarizes classification results on the hypothesis of one net fitting across all pilots, all days.

Table 5.2 Classification Results Across Pilots

| **Analysis** | **95% CI** | **Linear** | **Quadratic** | **MLP** |
|---|---|---|---|---|
| Initial 146 vars | Lower | **50.202** | N/A | 51.26 |
| | Mean | | | **51.61** |
| | Upper | | | 51.96 |
| SAS 59 vars | Lower | **48.79** | N/A | 51.57 |
| | Mean | | | **51.94** |
| | Upper | | | 52.31 |
| SNR 35 vars | Lower | **53.13** | N/A | 55.49 |
| | Mean | | | **55.78** |
| | Upper | | | 56.07 |
| Factor Analysis 18 vars | Lower | **52.12** | **55.45** | 55.26 |
| | Mean | | | **55.45** |
| | Upper | | | 55.64 |

Notice in Table 5.2 there are single values once again for the linear and the quadratic classifiers. The reason for these single numbers is the same as the reason given in the section above. In this case the entire pilot 1 data set is used to train the classifiers while the entire pilot 4 data set is used to test how well these classifiers perform.

### *5.3 Summary of Results*

After creating classifiers and testing the performance of these classifiers on individual pilots on a single day, the natural extension was to look at forming classifiers for two new scenarios: 1) investigate classifier performance on one pilot across two

days, and 2) investigate classifier performance across pilots, across days. The results are radically different from the results for classification on the individual pilots. In both scenarios, we barely get above 50% as a classification accuracy on the test sets. The highest CA measure for using pilot 1, day 1 to predict for pilot 1, day 2 was 60.68%. The highest CA measure obtained on the case of using all of pilot 1 to predict for pilot 4 was 55.78%. These results suggest that the classifiers are hardly better than just tossing a coin and guessing what the classification of a new exemplar will be.

The poor results of the classification efforts presented in this chapter raise questions as to why this happened. Chapter 6 extends some possible explanations and recommendations to fix this poor classification problem as well as recommendations for further research in the pilot mental workload arena.

# *VI. Conclusions and Recommendations*

This chapter summarizes the results of this research effort. Specifically, the results related to the different screening techniques used for feature selection and reduction are summarized, and a comparison is made between the results from the two screening methods. Additionally, results are summarized on a comparison of the modeling techniques used and how well each model performed as classification efforts moved from one pilot on one day, to multiple pilots over multiple days. Finally, recommendations for further research are presented.

## *6.1 Screening Techniques*

The initial data set, after all preprocessing was finished, contained 151 variables (146 in the case of pilot 4, day 1). This is a tremendous amount of variables to manage. Screening techniques were used to reduce the number of features required for classification. The *SAS* stepwise selection procedure produced a statistical method for determining the number of features that were required for classification. While the *SAS* procedure made an initial cut into the total number of input features required for classification, it tended to err on the conservative side. In all cases, the final number of input features determined to be the salient feature set was far less than the number of variables initially picked by the *SAS* stepwise procedure.

The second screening technique utilized was the SNR screening method. This method compared an injected noise feature to the features considered for input. The SNR screening method is a much more subjective method. The final number of features is determined by the researcher. The results from the SNR screening method gave more hope that comparable predictions could be made with an even smaller set of input variables as compared with the total input set or the *SAS* stepwise feature set. In every feature reduction effort, the number of input features selected with

the SNR screening method were less than the number chosen by the *SAS* stepwise screening method.

The input features chosen by the *SAS* stepwise procedure and the features chosen by the SNR screening method were not always the same for every data set presented to the two screening methods. The following question was asked, "Why are the screening methods selecting different variables?" It was proposed that factor analysis may provide some insight into this question. The factor analysis revealed that all of the significant EEG readings chosen by the screening methods were related to one of two factors. Variables in the central, frontal and occipital regions fell on factor 1. Variables from the parietal and temporal regions fell on factor 2. All peripheral measure fell on their own individual factors. For example, if heart rate, blinks, interblink interval and breaths were chosen as significant, heart rate and breaths would be loaded heavily on their own individual factors. Blinks and interblink interval would each be loaded on one common factor, since they are clearly related.

The factor analysis enabled us to see that even though the screening methods were choosing some different variables, the main factors inherent to the data set were being covered. A final cut was made on the number of input features based on this information. The final number of input features was based on the variables chosen by both the *SAS* stepwise procedure and the SNR screening method. Table 6.1 gives a quick summary on the reduction of features for each individual pilot for classification on one day.

Table 6.1 Factor Reduction

| **Pilot/Day** | **Initial** | **SAS** | **SNR** | **Factor Analysis** |
|---|---|---|---|---|
| Pilot 1/Day 1 | 151 | 34 | 14 | 10 |
| Pilot 1/Day 2 | 151 | 71 | 17 | 13 |
| Pilot 4/Day 1 | 146 | 79 | 5 | 3 |
| Pilot 4/Day 2 | 151 | 62 | 5 | 3 |

After the main factors driving the data were determined, an attempt was made to attach some meaning to these factors. As mentioned before, the first factor contained variables from the frontal, central and occipital regions. The frontal region is where all higher order planning takes place. The central region is where the brain controls all motor skills such as arm and leg movement. The occipital region is the area for vision. The first factor could be explained as a dimension related to higher planning and the motor skills used to carry out those plans and any visual information that the pilot is receiving. The second factor contained variables from the parietal and temporal regions. The dimension underlying factor 2 seems to indicate this factor is associated with the low level association processes of the brain and any auditory measures the pilot is receiving.

### *6.2 Comparison of Classification Models*

Three classifiers were used to predict pilot mental workload. In general, the neural networks were the best classifier. This became especially apparent when the number of input features was reduced. One problem that was encountered using the linear and quadratic classifiers was the instance where the covariance matrices were nearly singular. Initial inspection of the data revealed that many of the EEG readings were very highly correlated with other EEG readings. This correlation caused the covariance matrix to be nearly singular. This condition created enough problems that *Matlab* could not use the linear or quadratic classifier for prediction. In comparison, the neural network was able to perform every single time, regardless of how high the correlation was between variables considered for classification.

It was mentioned that the MLP was the best classifier especially when the number of input features was reduced. The linear classifier operates on the assumption that the covariance matrices of the two data sets are statistically equal. As the number of inputs are reduced, the chances the covariance matrices are equal begins to decline. This results in the linear classifier not predicting as well when

the number of input features are reduced. In one case, the classification accuracy of the linear predictor varied as much as 13% from the classification accuracy of the neural network. The quadratic classifier is a bit more flexible than the linear classifier. The linear classifier performs well when the data is somewhat similar (equal covariance structures) and the data sets linearly separable. The quadratic classifier is flexible in that it allows for unequal covariance structures and can adapt if the regions are not totally linear. However, when the structure of the inputs deviates from regions that can be separated by both the linear and quadratic classifier, the quadratic classifier performs poorly as well. The largest difference in classification accuracy between the quadratic classifier and an MLP was 4%. In the grand scheme of things, this is practically insignificant. This seems to suggest that the regions of interest, the regions that contain both groups, don't deviate wildly from an area that can be separated by a second order equation.

The discriminant models were both limited by the assumptions of the data structure and in certain cases, could not even produce a viable classifier if inputs were highly correlated. The MLP does not care about the structure of the input data. It is able to adapt to correlated data and extremely non-linear regions.

Classification accuracies of the data depend on what data set is being presented for classification. If we look strictly at classification accuracy of the MLPs, the individual classification accuracies for the pilots varied. Classification ranged from 97% to 74% for an individual pilot on a single day. While classification accuracy depends on the structure of the input data set, it also depends on the individual pilot being measured. Perhaps one pilot is more experienced than another. While readings respond to higher mental workload levels, they may not respond as drastically as a pilot that is less experienced, making classification for that pilot fall on the lower end.

An attempt was made to use one classifier formed for a single pilot on one day to predict for a second day of flight for the same pilot. Regardless of the

classification method used, the results were hardly better than flipping a coin and guessing which workload group an exemplar belonged to. The highest classification accuracy ever reached was 60%. An attempt was also made to use one pilot's data (both days) and predict workload level for a second pilot (both days). The results here were equally poor. The highest classification accuracy reached was 55%. The poor classification raises the question, "What is causing the problem?"

First, let's consider the scenario of trying to use one day to predict a second day. Heart rate was determined to be a common driving factor in all classification efforts. Therefore, the investigation focuses on the heart rate variable. Readings on heart rate were collected for the first and second days of flight. These readings were then plotted, as shown in Figure 6.1.

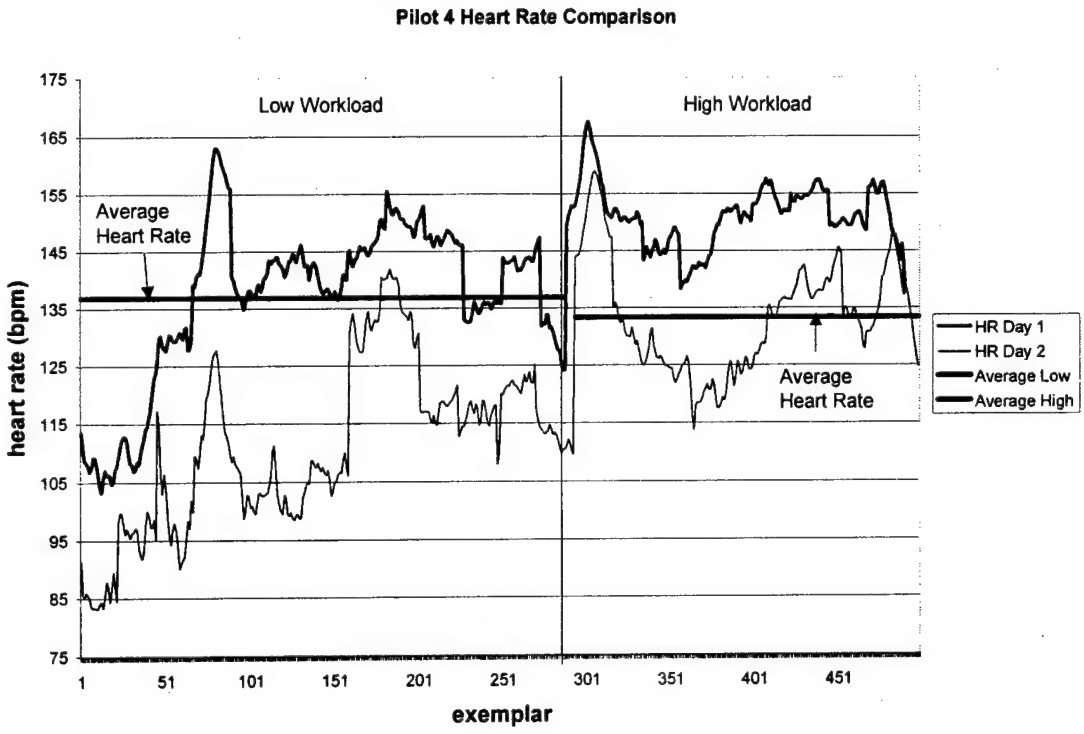


Figure 6.1 Comparison of Heart Rate for Pilot 4

The vertical line on the graph indicates where workload level changes from low to high. One immediate observation is possible. Overall, the readings for heart rate on day 2 are far lower than the readings for heart rate on day 1. There are two horizontal lines also represented in Figure 6.1. The horizontal line in the *low* workload segment indicates the average heart rate for the low workload segments on day 1. This average was 136.9 beats per minute. The second horizontal line indicates the average heart rate for the *high* workload segments of flight on the second day. As the figure shows, this average is less than the average for the low workload segments on the first day, at a value of 133.4 bpm. This introduces an interesting dilemma. The network and statistical classifiers were all trained on day one to predict for day two. Since the average of the high workload heart rate readings on day 2 are is less than the average of the low workload heart rate readings on day 1, almost all of the exemplars from day 2 will be classified as low workload. This will lead to about a 50% classification accuracy because more than half of the flight is actually at a low workload.

A similar investigation was done into the classification across pilots. Figure 6.2 shows a pictorial view of the same dilemma. Recall that pilot 1 data was used to form the classifiers to predict mental workload level for pilot 4. Notice there are two horizontal lines in Figure 6.2. The first horizontal line is the average heart rate of the low workload level for pilot 4, with an average of 123.8 beats per minute. The second horizontal line is the average heart rate of the high workload level for pilot 1, with an average of 106.1 beats per minute. This indicates that almost every exemplar for pilot 4 will classify as a high workload level. Once again, since portions of the flight are indeed at a high workload level, about half of all classifications will be correct. The trends that we see in trying to classify across days or across pilots have supported the conclusion that different people act differently and people act differently on different days. Somehow these differences must be compensated for before any useful classification efforts can be made.

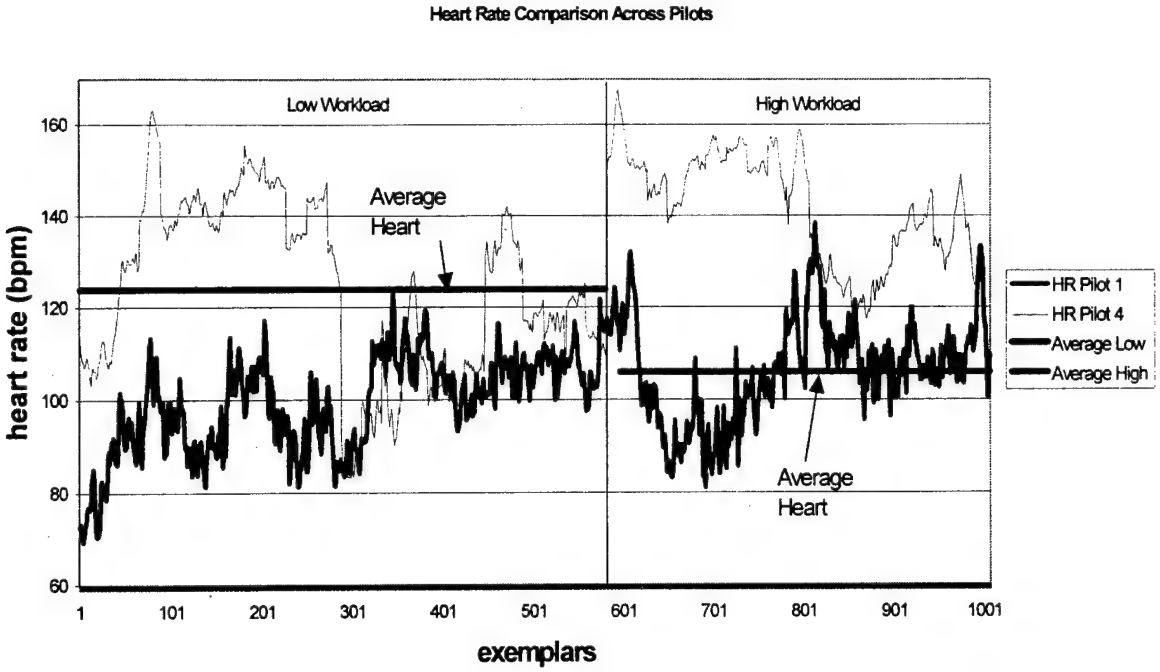


Figure 6.2 Comparison of Heart Rate Across Pilots

## *6.3 Recommendations*

There are several opportunities for further research on the subject presented in this thesis. These recommendations are made with the idea that someday a system could be put into a cockpit be put to practical use to save a pilot's life.

*6.3.1 Recurrent Neural Networks.* The EEG and peripheral measures that were collected are all collected over time. For this research effort, this time dependency was removed (via a fast-Fourier transform) and classifications were made solely on the frequency based EEG readings. Recurrent neural networks (RNN) have the ability to adapt to time dependent inputs. A recurrent neural network is different from a feedforward neural network in that as it trains, it uses the outputs from each epoch as inputs to the next epoch. The RNN uses past information to make decisions about future classification. The introduction of recurrent neural net-

works may lead to a more precise and perhaps higher classification accuracy. Some research on using recurrent neural networks to predict pilot mental workload was done in Greene's dissertation [11].

*6.3.2 Batch Means.* The data used in this research was subject to a large amount of preprocessing. As mentioned above, the raw EEG passed through a FFT. The FFT took out the time dependency and passed on a frequency based signal. The output was a frequency of 1-256 Hz for each second of data. The frequency signal was then filtered and the power was collected at five frequency bands. The power collected at each second was then averaged over a ten second interval. Some overlap was included in these power estimates in order to smooth out the data readings. By doing this, each ten second window is highly correlated with the next ten second window. One of the underlying assumptions for the classification models is that the data is independent. This assumption is clearly suspect early in the classification process. When we look at the classification results of the statistical classifiers compared to the classification results of the neural networks the violation of these assumptions did not seem to make much of a difference. Laine [14] used this method and classification did not seem to suffer; he frequently classified data at 100%. Classification accuracy did not seem to suffer that much in this research effort either. In one case, classification for pilot 4 on day one was as high as 97%. The question arises, however, about possible classification improvements using data that is not correlated. This suggests using batch means to calculate the average power estimates. There are several suggestions and algorithms that indicate what batch size to use. A possible result could be to average power readings using a batch size of 12 seconds. Of course, no overlap is included using the method of batch means.

*6.3.3 Classification Across Days or Across Pilots.* The Air Force would like to implement some type of warning system into a cockpit to prevent fatalities.

In order for this idea to be practical two things have to happen. The classifier has to be almost 100% accurate 100% of the time and the classifier has to be practical. A classifier that has to be retrained every flight or a classifier that won't work for different pilots is not very practical. It was observed that classification across days or across pilots does not seem feasible. An investigation into the structure of the data showed that pilots react very differently from day to day and react differently compared to other pilots. For example, in Figure 6.2 we can see that, on average, pilot 4 has a much higher heart rate than pilot 1. These differences led to classification that was little better than flipping a coin to classify exemplars. Right now, the code written does not take into account this bias that is present in the data presented for classification If a way could be found to account for any bias that may be present in the data, prediction from day to day or from pilot to pilot looks feasible.

# *Appendix A. Flight Segments and Associated Workload Level*

Table A.1 Flight Segments

| Flight Segment | Workload Level |
|---|---|
| Baseline 1 | 1 |
| Preflight | 1 |
| Engine Start | 1 |
| VFR Takeoff | 1 |
| VFR Climbout 1 | 1 |
| VFR Cruise | 1 |
| VFR Airwork | 1 |
| Approach | 1 |
| VFR Touch and Go | 2 |
| VFR Climbout 2 | 1 |
| IFR Airwork | 2 |
| IFR Cruise | 2 |
| IFR Hold | 2 |
| IFR DME Arc | 2 |
| IFR ILS Tracking | 2 |
| IFR Missed Approach | 2 |
| IFR Climbout | 1 |
| HS Hold | 1 |
| HS DME Arc | 1 |
| HS ILS Tracking | 2 |
| Landing | 2 |
| Baseline 2 | 1 |

**VFR** Visual Flight

**IFR** Instrument Flight

**HS** High Speed

**DME** Distance Measuring Equipment

**ILS** Instrument Landing System

# *Appendix B. Pilot Subjective Measures of Mental Workload*

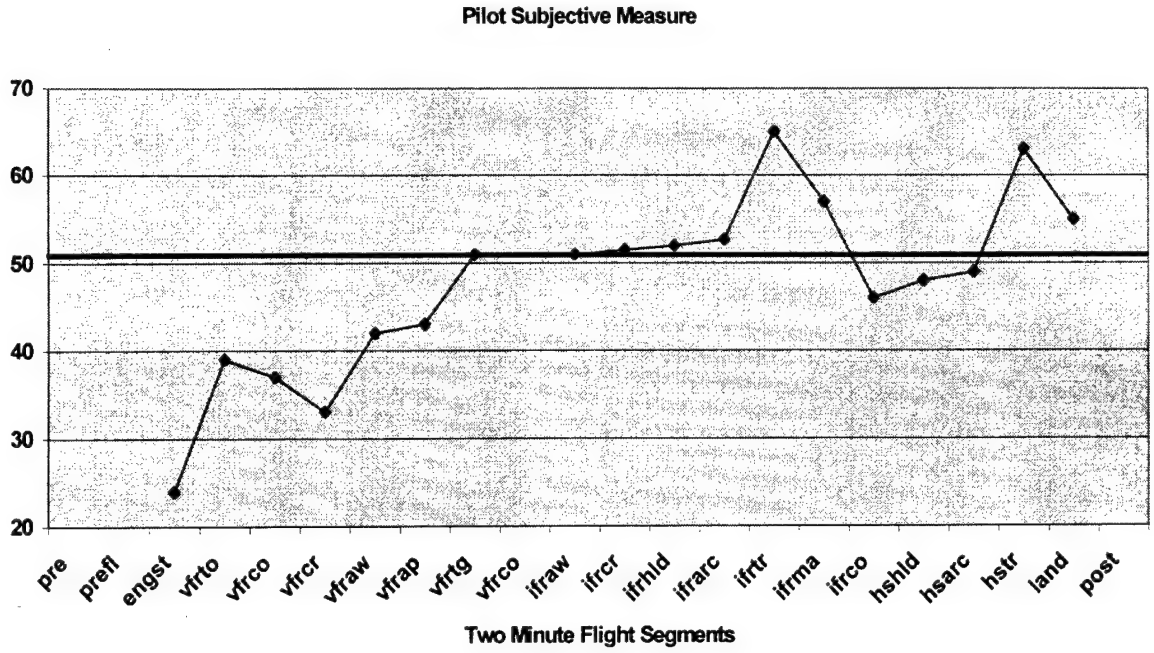


Figure B.1 Pilot Subjective Measure

## *Appendix C. Fortran Formatting Code*

```
Program makeit
      real x(151)
      do 10 ii=1,506
      read(1,*)(x(i),i=1,151),ispec
      write(2,*)(x(j),j=1,151),ispec
10    continue
      end
```

# *Appendix D. Variables Used for Classification After Screening*

Table D.1 Pilot 1, Day 2 SAS Screening Results

| **Variable** | **Variable** | **Variable** | **Variable** | **Variable** | **Variable** |
|---|---|---|---|---|---|
| C3b | CZa | F8d | FP2d | OZt | P9a |
| C3ub | CZb | FC1d | FP2a | OZa | P9b |
| C4a | CZub | FC1b | FP2ub | P10d | PO3t |
| C5t | F3d | FC2d | FZub | P10b | PO3b |
| C5b | F3a | FC2t | IZd | P10ub | PO3ub |
| C5ub | F3ub | FC2b | IZt | P3t | PO4d |
| C6d | F4d | FC2ub | IZub | P3a | PO4t |
| C6t | F4a | FP1d | O2d | P3ub | PO4a |
| C6b | F4b | FP1t | O2t | P4t | PO4b |
| CZd | F7a | FP1b | O2a | P8ub | PZt |
| CZt | F7b | FP1ub | OZd | P9t | PZub |
| T8d | T8b | HR | BLNKS | BRTHS | |

Table D.2 Pilot 1, Day 2 SNR Screening Results

| **Variable** | **Variable** | **Variable** |
|---|---|---|
| C5ub | FP2a | P8ub |
| CZub | IZt | PO3a |
| F3d | IZb | PO4t |
| FC2t | OZd | PZt |
| FP2d | P10t | HR |
| BLNKS | IBRI | |

Table D.3 Pilot 1, Day 2 Factor Analysis Results

| Variable | Variable | Variable |
|---|---|---|
| C5ub | FC2t | IZt |
| CZub | FP2d | OZd |
| F3d | FP2a | P8ub |
| PO4t | PZt | HR |
| BLNKS | | |

Table D.4 Pilot 4, Day 1 SAS Screening Results

| Variable | Variable | Variable | Variable | Variable | Variable | Variable |
|---|---|---|---|---|---|---|
| C3t | CZt | F8d | FZd | P10a | PO3d | T7t |
| C3b | CZb | FC1t | O1d | P3d | PO3a | T7a |
| C3ub | CZub | FC1a | O1t | P3t | PO3b | T7b |
| C4t | F3d | FC1ub | O1ub | P3b | PO4d | T7ub |
| C4b | F3t | FC2d | O2d | P3ub | PO4a | T8d |
| C5t | F3a | FC2ub | O2t | P4d | PO4b | T8a |
| C5ub | F3b | FP1d | O2b | P4a | PO4ub | T8b |
| C6d | F3ub | FP1b | OZd | P4b | PZt | T8ub |
| C6t | F4d | FP1ub | OZt | P7b | PZa | HR |
| C6a | F4ub | FP2d | OZb | P9t | PZb | BLNKS |
| C6ub | F7d | FP2b | P10d | P9a | T7d | IBLI |
| IBRI | | | | | | |

Table D.5 Pilot 4, Day 1 SNR Screening Results

| Variable | Variable | Variable |
|---|---|---|
| CZd | FP2b | HR |
| CZa | T7a | |

Table D.6 Pilot 4, Day 1 Factor Analysis Results

| Variable |
|---|
| FP2b |
| T7a |
| HR |

Table D.7 Pilot 4, Day 2 SAS Screening Results

| Variable | Variable | Variable | Variable | Variable | Variable |
|---|---|---|---|---|---|
| C3ub | F3b | F8a | O1d | P7b | Pzd |
| C4d | F4d | F8ub | O2d | P7ub | PZt |
| C4b | F4a | FC1d | O2t | P9b | PZub |
| C5a | F4b | FC1t | O2a | P9ub | T7d |
| C5b | F7d | FP1t | O2b | PO3a | T7a |
| C6a | F7t | FP1ub | OZt | PO3ub | T7ub |
| C6b | F7a | FP2d | OZb | PO4d | T8d |
| C6ub | F7ub | FP2t | OZub | PO4t | T8a |
| Czd | F8d | FZd | P3d | PO4a | T8ub |
| Czub | F8t | IZt | P3ub | PO4ub | HR |
| IBI | BRTHS | | | | |

Table D.8 Pilot 4, Day 2 SNR Screening Results

| Variable | Variable | Variable |
|---|---|---|
| O2d | P8d | HR |
| P3d | PO3t | |

Table D.9 Pilot 4, Day 2 Factor Analysis Results

| Variable |
|---|
| O2d |
| P3d |
| HR |

# *Appendix E. Factor Loadings for Individual Pilots*

| | Factor 1 | Factor 2 | Factor 3 | Factor 4 | Factor 5 | Factor 6 | Factor 7 | Factor 9 | Factor 10 | Factor 11 | Factor 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C3t | | | 0.88289 | | | | | | | | |
| C3b | | | 0.89014 | | | | | | | | |
| C3ub | | | 0.78961 | | | | | | | | |
| C4t | | | 0.82782 | | | | | | | | |
| C4b | | | 0.43504 | 0.69179 | | | | | | | |
| C4t | -0.73287 | | | | | | | | | | |
| C4ub | | | 0.53886 | | | | | | | | |
| C6d | | | 0.5095 | | | | | | 0.54147 | | |
| C6t | | | 0.63525 | | | | | | | | |
| C6a | | | 0.5615 | | | | 0.4425 | | | | |
| C6ub | | | 0.63755 | | | | | | | | |
| CZd | | | 0.54035* | | | | | | | | 0.60935* |
| CZt | | | 0.65328 | | | | | | | | |
| CZa | | | | | | | | | | | 0.66474* |
| CZb | | | | | | 0.79695 | | | | | |
| CZub | | | 0.61711 | | | | | | | | |
| F3d | | | 0.82827 | | | | | | | | |
| F3t | | | 0.71455 | | | | | | | | |
| F3a | | | 0.88086 | | | | | | | | |
| F3b | -0.83082 | | | | | | | | | | |
| F3ub | | | | | 0.9214 | | | | | | |
| F4d | | 0.89144 | | | | | | | | | |
| F4ub | | 0.74941 | | | | | | | | | |
| F7d | | 0.8934 | | | | | | | | | |
| F8d | | 0.86274 | | | | | | | | | |
| FC1t | | 0.85015 | | | | | | | | | |
| FC1a | | 0.83826 | | | | | | | | | |
| FC1ub | | 0.80587 | | | | | | | | | |
| FC2d | | 0.79298 | | | | | | | | | |
| FC2ub | | 0.90844 | | | | | | | | | |
| FP1d | | 0.83233 | | | | | | | | | |
| FP1b | | | | | 0.95329 | | | | | | |
| FP1ub | | 0.86177 | | | | | | | | | |
| FP2d | | 0.7599 | | | | | | | | | |
| FP2b | | 0.82022 | | | | | | | | | |
| FZd | | | | -0.78053 | | | | | | | |
| O1d | | 0.74217 | | | | | | | | | |
| O1t | | 0.73516 | | | | | | | | | |
| O1ub | 0.59576 | 0.66707 | | | | | | | | | |
| O2d | | 0.71155 | | | | | | | | | |
| O2t | | 0.67888 | | | | | | | | | |
| O2b | | 0.86316 | | | | | | | | | |
| OZd | | 0.77888 | | | | | | | | | |
| OZt | -0.63479 | | | | | | | | | | |
| OZb | 0.88345 | | | | | | | | | | |
| OZub | 0.92957 | | | | | | | | | | |
| P10d | 0.92733 | | | | | | | | | | |
| P10a | 0.87185 | | | | | | | | | | |
| P3d | 0.84323 | | | | | | | | | | |
| P3t | 0.88949 | | | | | | | | | | |
| P3b | 0.77325 | | | | | | | | | | |
| P3ub | 0.88584 | | | | | | | | | | |
| P4d | 0.62874 | 0.51692 | | | | | | | | | |
| P4a | 0.87123 | | | | | | | | | | |
| P4b | 0.83886 | | | | | | | | | | |
| P7b | 0.90982 | | | | | | | | | | |
| P9t | 0.74081 | | | | | | | | | | |
| P9a | 0.91474 | | | | | | | | | | |
| PO3d | 0.95021 | | | | | | | | | | |
| PO3a | 0.90785 | | | | | | | | | | |
| PO3b | 0.86733 | | | | | | | | | | |
| PO4d | 0.92328 | | | | | | | | | | |
| PO4a | 0.79481 | | | | | | | | | | |
| PO4b | 0.90473 | | | | | | | | | | |
| PO4ub | 0.66907 | | | | | | | | | | |
| PZt | 0.87325 | | | | | | | | | | |
| Pza | 0.85368 | | | | | | | | | | |
| PZb | 0.89026 | | | | | | | | | | |
| T7d | 0.9164 | | | | | | | | | | |
| T7t | 0.8352 | | | | | | | | | | |
| T7a | 0.91511 | | | | | | | | | | |

Figure E.1 Factor Analysis on Pilot 4, Day 1

| | Factor 1 | Factor 2 | Factor 3 | Factor 4 | Factor 6 | Factor 7 | Factor 9 |
|---|---|---|---|---|---|---|---|
| C3b | 0.90641 | | | | | | |
| C3ub | 0.90502 | | | | | | |
| C4a | 0.89246 | | | | | | |
| C5t | 0.73597 | | | | | | |
| C5b | 0.88363 | | | | | | |
| C5ub | **0.85669** | | | | | | |
| C6d | 0.93247 | | | | | | |
| C6t | 0.93434 | | | | | | |
| C6b | 0.91621 | | | | | | |
| CZd | 0.91646 | | | | | | |
| CZt | 0.91552 | | | | | | |
| CZa | 0.90639 | | | | | | |
| CZb | 0.92065 | | | | | | |
| Czub | **0.82498** | | | | | | |
| F3d | **0.77815** | | | | | | |
| F3a | 0.91923 | | | | | | |
| F3ub | | | 0.97065 | | | | |
| F4d | 0.91827 | | | | | | |
| F4a | 0.92277 | | | | | | |
| F4b | 0.90056 | | | | | | |
| F7a | 0.85758 | | | | | | |
| F7b | 0.90513 | | | | | | |
| F8d | 0.79918 | | | | | | |
| FC1d | 0.93787 | | | | | | |
| FC1b | 0.90135 | | | | | | |
| FC2d | 0.91693 | | | | | | |
| FC2t | **0.93966** | | | | | | |
| FC2b | 0.90039 | | | | | | |
| FC2ub | 0.90374 | | | | | | |
| FP1d | 0.90783 | | | | | | |
| FP1t | 0.93594 | | | | | | |
| FP1b | 0.97136 | | | | | | |
| FP1ub | 0.91627 | | | | | | |
| FP2d | **0.90541** | | | | | | |
| FP2a | **0.90361** | | | | | | |
| FP2ub | 0.91015 | | | | | | |
| FZub | 0.81317 | | | | | | |
| IZd | 0.79265 | | | | | | |
| IZt | **0.90551** | | | | | | |
| IZb | 0.92763* | | | | | | |
| IZub | 0.9332 | | | | | | |
| O2d | 0.93689 | | | | | | |
| O2t | 0.88097 | | | | | | |
| O2a | 0.91059 | | | | | | |
| OZd | **0.92567** | | | | | | |
| OZt | 0.87594 | | | | | | |
| OZa | | | 0.93058 | | | | |
| P10d | 0.73516 | | | | | | |
| P10t | 0.72457* | | | | | | |
| P10b | 0.74711 | | | | | | |
| P10ub | 0.75502 | | | | | | |
| P3t | 0.76228 | | | | | | |
| P3a | 0.76019 | | | | | | |
| P3ub | 0.69776 | | | 0.57834 | | | |
| P4t | 0.73045 | | | | | | |
| P8ub | 0.73281 | | | | | | |
| P9t | | | 0.88164 | | | | |
| P9a | | 0.87372 | | | | | |
| P9b | | 0.90728 | | | | | |
| PO3t | | 0.88487 | | | | | |
| PO3a | | 0.86546* | | | | | |
| PO3b | | 0.90145 | | | | | |
| PO3ub | | 0.89032 | | | | | |
| PO4d | | 0.87122 | | | | | |
| PO4t | | **0.86057** | | | | | |
| PO4a | | 0.61247 | | 0.61401 | | | |
| PO4b | | 0.57193 | | 0.67212 | | | |
| PZt | | **0.85001** | | | | | |
| Pzub | | 0.89851 | | | | | |
| T8d | | 0.8775 | | | | | |
| T8b | | 0.90539 | | | | | |

Figure E.2 Factor Analysis on Pilot 1, Day 2

| | Factor 1 | Factor 2 | Factor 3 | Factor 4 | Factor 5 | Factor 7 | Factor 9 | Factor 11 | Factor 13 | Factor 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| C3ub | | | 0.90377 | | | | | | | |
| C4d | | | 0.8874 | | | | | | | |
| C4b | | | 0.67132 | | | | | | | |
| C5a | | | | -0.67954 | | | | | | |
| C5b | | | | 0.87785 | | | | | | |
| C6a | | | | | 0.75065 | | | | | |
| C6b | | | | | | | | | | 0.55387 |
| C6ub | | | | | 0.66418 | | | | | |
| CZd | | | | | 0.7259 | | | | | |
| CZub | | | | | 0.45315 | | | | | 0.41048 |
| F3b | -0.70469 | | | | | | | | | |
| F4d | 0.64197 | | | | | | | | | |
| F4a | 0.64626 | | | | | | | | | |
| F4b | 0.63777 | | | | | | | | | |
| F7d | 0.66981 | | | | | | | | | |
| F7t | 0.67746 | | | | | | | | | |
| F7a | 0.67483 | | | | | | | | | |
| F7ub | 0.66477 | | | | | | | | | |
| F8d | 0.55272 | | | | | | | 0.48928 | | |
| F8t | 0.61464 | | | | | | | | | |
| F8a | | | | 0.94699 | | | | | | |
| F8ub | 0.73503 | | | | | | | | | |
| FC1d | 0.69537 | | | | | | | | | |
| FC1t | 0.67898 | | | | | | | | | |
| FP1t | 0.76401 | | | | | | | | | |
| FP1ub | | 0.84346 | | | | | | | | |
| FP2d | | 0.65278 | | | | | | | | |
| FP2t | | 0.76573 | | | | | | | | |
| FZd | 0.59467 | 0.53435 | | | | | | | | |
| IZt | | | | 0.87857 | | | | | | |
| O1d | | 0.65613 | | | | | | | | |
| O2d | | 0.68779 | | | | | | | | |
| O2t | | 0.71047 | | | | | | | | |
| O2a | | 0.76346 | | | | | | | | |
| O2b | | 0.8828 | | | | | | | | |
| OZt | | | | | | 0.66082 | | | | |
| OZb | 0.92588 | | | | | | | | | |
| OZub | 0.88969 | | | | | | | | | |
| P3d | 0.89629 | | | | | | | | | |
| P3ub | 0.84694 | | | | | | | | | |
| P7b | 0.92995 | | | | | | | | | |
| P7ub | 0.93832 | | | | | | | | | |
| P8d | 0.91456* | | | | | | | | | |
| P9b | 0.938 | | | | | | | | | |
| P9ub | 0.90633 | | | | | | | | | |
| PO3t | 0.94196* | | | | | | | | | |
| PO3a | 0.92315 | | | | | | | | | |
| PO3ub | 0.89394 | | | | | | | | | |
| PO4d | 0.94631 | | | | | | | | | |
| PO4t | 0.85458 | | | | | | | | | |
| PO4ub | | | | 0.70966 | | | | | | |
| PZd | 0.85551 | | | | | | | | | |
| PZt | 0.88078 | | | | | | | | | |
| Pzub | 0.91228 | | | | | | | | | |
| T7d | 0.94277 | | | | | | | | | |
| T7a | 0.91492 | | | | | | | | | |
| T7ub | 0.90694 | | | | | | | | | |
| T8d | 0.94948 | | | | | | | | | |
| T8a | 0.92989 | | | | | | | | | |
| T8ub | 0.52634 | 0.40015 | | | | | | | | |
| HR | | | | | | | | | 0.66211 | |
| IBI | | | | | | | | | 0.7607 | |
| BRTHS | | | | | | | 0.86036 | | | |

Figure E.3 Factor Analysis for Pilot 4, Day 2

# *Bibliography*


1. Air Force Research Laboratory, AFRL, "Flight Psychophysiology Laboratory," 1998. Office Brochure, Flight Psychophysiology Laboratory, Human Interface Technology Branch, Crew System Interface Division, Human Effectiveness Directorate (AFRL/HECP).
2. Auten, J. "G-LOC: Is the Cluebag Half Full or Half Empty?," *Flying Safety*, *52*:5–6 (1996).
3. Bauer, K.W., "OPER685, Applied Multivariate Data Analysis," Fall 1999. Air Force Institute of Technology, OH.
4. Bauer, K.W., et al. "Feature Screening using Signal-to-Noise Ratios," *Neurocomputing* accepted April 29, 1999.
5. Belue, L.M. and K.W. Bauer. "Determining Input Features for Multilayer Perceptrons," *Neurocomputing*, *7*:111–121 (1995).
6. Bishop, C.M. *From Neural Networks for Pattern Recognition.* Oxford, UK: Clarendon Press, 1996. 38.
7. Burden, R.L. and J.D. Faires. *Numerical Analysis* (Fifth Edition). Boston, MA: PWS Publishing Co., 1993.
8. Cox, K.S. *An Analysis of Noise Reduction Using Backpropagation Neural Networks.* MS thesis, School of Engineering, December.
9. Damos, D.L, editor. *Multiple-task Performance.* London: Taylor and Francis Ltd., 1996. 328–360.
10. Dillon, W.R. and M. Goldstein, editors. *Multivariate Analysis: Methods and Applications.* New York: John Wiley and Sons, Inc., 1984. 360–393.
11. Greene, K.A. *A Feature Saliency in Artificial Neural Networks with Application to Modeling Workload.* PhD dissertation, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1998.
12. Griffiths, D.J. *Intoduction to Electrodynamics* (Second Edition). Englewood Cliffs, NJ: Prentice Hall, 1989. 346–350.
13. Hankins, T.C. and G.F. Wilson. "A Comparison of Heart Rate, Eye Activity, EEG and Subjective Measures of Pilot Mental Workload during Flight," *Aviation, Space, and Environmental Medicine* April.

14. Laine, T.I. *Selection of Psychophysiological Features Across Subjects for Classifying Workload Using Artificial Neural Networks*. MS thesis, School of Engineering, March.

15. Levy-Leblond, Jean-Marc and F. Balibar. *Quantics - Rudiments of Quantum Physics*. Amsterdam, NY: Elsevier SCience Publishing Company, 1990. 42–50.

16. MathWorks, Inc. *MATLAB Signal Processing Toolbox User's Guide*. Natick, MA: MathWorks, 1998. 2-2, 3-5–3-9.

17. McCulloch, W.S. *Embodiments of Mind*. Cambridge, MA: MIT Press, 1988.

18. Smith, M. *Neural Networks for Statistical Modeling*. Boston: International Thomson Computer Press, 1996.

19. Steppe, J.M. and K.W. Bauer. "Improved Feature Screening in Feedforward Neural Networks," *Neurocomputing*, *13*:47–58 (1996).

20. Steppe, J.M., et al. "Integrated Feature and Architecture Selection," *IEEE Transactions on Neural Networks*, 7(4):1007–1014 (July 1996).

21. Sumrell, D.B. *An Investigation of Preliminary Feature Screening Using Dignal-to-Noise Ratios*. MS thesis, School of Engineering, March.

22. Swingler, K. *Applying Neural Networks: A Practical Guide*. San Diego, CA: Academic Press, 1996.

23. The American Heritage College Dictionary, Third Edition, "physiological." Houghton Mifflin Company, Boston, 1993.

24. The American Heritage College Dictionary, Third Edition, "psycho." Houghton Mifflin Company, Boston, 1993.

25. Wackerly, D.D., et al. *Mathematical Statistics with Applications*. Belmont, CA: Wadsworth Publishing Co., 1996. 224.

26. Wasserman, P.D. *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold, 1989.

27. Weiss, S.M. and C.A. Kulikowski. *Computer System that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1991.

28. Wilson, G.F. "Applied Use of Cardiac and Respiration Measures: Practical Considerations and Precautions," *Biological Psychology*, *34*:163–178 (1992).

29. Wilson, G.F. "Air-to-Ground Training Missions: A Psychophysiological Workload Analysis," *Ergonomics*, *36*(9):1071–1087 (1993).

30. Wilson, G.F. and F. Fisher. "Cognitive Task Classification Based Upon Topographical EEG Data," *Biological Psychology*, *40*:239–250 (1995).

31. Wilson, G.F. and F. Fisher. "The Use of Cardiac and Eye Blink Measures to Determine Flight Segment in F4 Crews," *Aviation, Space, and Environmental Medicine*, *33*:959–962 (October 1997).

32. Wilson, G.F., et al. "Evoked Potential, Cardiac, Blink, and Respiration Measures of Pilot Workload in Air-to-Ground Missions," *Aviation, Space, and Environmental Medicine* February.

## *Vita*

Lt Julia East was born on 29 Jan 1976 in Oakland, CA. She graduated from Orestimba High School, Newman, CA in 1994 and was accepted to the US Air Force Academy, class of 1998. Lt. East graduated from the Academy in 1998 with a bachelor of science in mathematics. Her first assignment directly out of the Academy was to continue on in the academic world at the Air Force Institute of Technology. After graduation from AFIT, Lt East will be heading to the Air Force Personnel Center analyst shop at Randolph AFB, TX.